

Preprocesamiento para Ciencias de los Datos

M. Sc. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Computación, bachillerato en Ingeniería en Computación,
PAttern Recongition and MACHine Learning Group (PARMA-Group)

26 de junio de 2019

Resumen

Este material está basado en el libro *Análisis de señales*, de Pablo Irarrazabal[2], y el libro de Procesamiento digital de imágenes, de Rafael González, [1].

1 Preprocesamiento

El preprocesamiento consiste en aplicar a un conjunto de datos, representado matricialmente $X \in \mathbb{R}^{n \times m}$, una transformación:

$$X_p = T(X)$$

la cual **modifique la información en cada observación** o muestra $\vec{x}_i \in \mathbb{R}^n$, y/o **modifique la cantidad de muestras** m con el fin de mejorar el desempeño de etapas posteriores. Es por ello que entonces $X_p \in \mathbb{R}^{n \times m'}$. Observe que las técnicas de preprocesamiento preservan la dimensionalidad original de las muestras $\vec{x}_i \in \mathbb{R}^n$.

2 Preprocesamiento de registros

Muchas veces los datos a utilizar para resolver distintas preguntas en las ciencias de los datos, se conocen como **registros**, **tuplas** o **vectores de atributos**, los cuales hasta ahora hemos representado como arreglos $\vec{x} \in \mathbb{R}^n$:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

donde entonces, cada componente, para un registro, es denominado como un **atributo** (también a veces conocido como una dimensión, característica o variable). Un atributo x_i puede tomar un valor discreto o continuo. La distribución

de un atributo o variable se le conoce como distribución univariable, y de dos atributos, bivariable, etc.

Los siguientes son algunos tipos básicos de atributos:

1. **Atributos numéricos** $x_i \in \mathbb{N}$ o $x_i \in \mathbb{R}$: Los atributos numéricos son atributos que toman una serie de valores discretos o continuos. Las operaciones matemáticas están definidas en estos atributos, tal como la moda, media y mediana y las **distancias** (ℓ_p por ejemplo). Por ejemplo atributos como *temperatura*, *humedad* o *fecha* pueden tomar infinidad de valores, por lo que son entonces **continuos**. Atributos numéricos **discretos** son aquellos que toman valores naturales, como por ejemplo *cantidad_clientes*.
2. **Atributos ordinales** $x_i \in \mathbb{N}$: Corresponden a atributos con un conjunto de valores los cuales pueden ser ordenados de forma significativa, aunque las magnitudes de tales valores no sea conocida.
 - (a) Por ejemplo atributos como *satisfaccion_usuario* el cual puede tomar los valores *muy_satisfecho*, *poco_satisfecho*, o *insatisfecho*, a los cuales se les pueden asignar los valores 3,2 y 1 respectivamente.
 - (b) Los atributos ordinales pueden derivarse a partir de un atributo numérico, partiendo el dominio de tal atributo en intervalos, como por ejemplo el atributo *temperatura* puede tomar los valores 0-15, 16-25, 25-50.
 - (c) Para los atributos ordinales se puede calcular la moda (valor más frecuente), o mediana, con significado matemático, no así la media.
3. **Atributos binarios** $x_i \in \{0,1\}$: Un atributo binario es un atributo cualitativo que puede tomar únicamente dos valores (Booleano), usualmente codificados como 0 o 1. Un ejemplo de atributo binario es *identificacion_nacional* el cual es 0 si la identificación es extranjera, y 1 de lo contrario. Existen dos tipos de atributos binarios:
 - (a) **Atributos binarios simétricos**: son atributos cuyos valores presentan usualmente igual probabilidad. Por ejemplo el atributo de *genero* puede ser *hombre* o *mujer* con igual probabilidad usualmente.
 - (b) **Atributos binarios asimétricos**: se refiere así a los atributos cuyos valores pueden presentar probabilidades distintas. Por ejemplo, la variable *hiv* puede tomar los valores *negativo* = 0 o *positivo* *negativo* = 1, con lo que es usual que el valor de *negativo* sea más frecuente, con lo cual se usa asignarle el valor numérico de 0 al valor de mayor frecuencia.
4. **Atributos nominales o categóricos** $x_i \in \mathbb{N}$: son atributos cualitativos que se refieren a símbolos o nombres de elementos. Cada valor representa algún tipo de categoría, código o estado, con valores sin ningún tipo de relación de precedencia. También se les conoce como enumeraciones.

Valor de nacionalidad	<i>variable_dummy1</i>	<i>variable_dummy2</i>	<i>variable_dummy3</i>
1: Armenio	1	0	0
2: Ucraniano	0	1	0
3: Bielorruso	0	0	1
4: Ruso (variable de ref.)	0	0	0

Cuadro 1: Ejemplo de registros con atributo faltante.

- (a) Por ejemplo atributos como la *contextura_cuerpo* puede presentar distintos valores: *delgado*, *grueso*, *muy delgado*, etc. Otro ejemplo de atributo categórico el *estado_civil* el cual puede tomar los valores de *soltero*, *casado* o *divorciado*.

Observe que aunque los distintos valores que puede tomar una variable categórica son numéricos (para la variable *estado_civil* se puede asignar *soltero* = 0, *casado* = 1, etc.), al **no existir relación de precedencia entre los valores, estos no pueden operarse matemáticamente**. Por ejemplo, no tiene sentido calcular la media o la mediana de atributos categóricos. Además, para calcular la distancia entre dos vectores de atributos nominales $\vec{v}, \vec{w} \in \mathbb{R}^n$, se deben utilizar otras distancias como la **tasa de diferencias**:

$$d(\vec{v}, \vec{w}) = \frac{n - p}{n}$$

donde p se refiere a la cantidad de atributos nominales distintos.

Es por ello que el manejo y representación de atributos nominales debe ser distinto, como se detallará más adelante.

2.1 Codificación de variables categóricas

Las variables categóricas al no ser posible manejarse directamente de forma numérica, deben traducirse a una representación numérica. Tómese el siguiente ejemplo en el cual se toma una variable nominal *nacionalidad*, la cual puede tomar tres valores: ruso, ucraniano, armenio y bielorruso, en este caso entonces con $n = 4$ niveles. A continuación se presenta el enfoque de **codificación tonta o dummy**

La codificación tonta o *dummy* crea una variable dicotómica o binaria por los $n - 1$ valores, y para el otro valor restante, conocido como el valor de referencia, el cual se codifica con todas las otras variables dicotómicas en cero. La tabla 1 muestra la codificación tonta para la variable categórica o nominal *nacionalidad*.

El valor de referencia puede escogerse usando un criterio estadístico de las pruebas, definiendo el valor de referencia como el modelo base, con ausencia de aplicación de un *tratamiento*.

Ejemplo: Para entender mejor el efecto de la codificación *dummy*, tomese el siguiente problema: estimar el nivel de felicidad usando dos variables:

Nombre	Edad	Razon_social
Alejandro Bertinelli	15	CUIT
Luana Grifo	–	CUIL
Betina Roca	12	CUIT

Cuadro 2: Ejemplo de registros con atributo faltante.

estimar la felicidad de una persona $y \in [0 - 100]$ usando los siguientes predictores o variables de entrada:

- Salario neto $x_1 \in \mathbb{R}$, variable numérica, con valores positivos en dólares.
- Religión $x_2 \in \mathbb{N}$: variable categórica o binaria, con dos niveles: ateo y religioso. Lo modelaremos con una sola variable *dummy* x_2

El modelo de regresión lineal a construir viene entonces dada por:

$$y(\vec{x}) = w_0 + w_1x_1 + w_2x_2$$

Observe que el modelo puede simplificarse de la siguiente forma:

$$y(\vec{x}) = \begin{cases} \beta_0 + w_1x_1 & \text{con } x_2 = 1 \text{ y } \beta_0 = w_0 + w_2 \\ w_0 + w_1x_1 & \text{con } x_2 = 0 \end{cases}$$

lo cual indica que la regresión realiza la estimación de dos modelos distintos para las muestras con variable categórica $x_2 = 0$ y $x_2 = 1$, correspondientes a dos líneas paralelas. Un modelo indicará el efecto del tratamiento *religión*, y el otro indicará la ausencia de este.

Existen otros tipos de codificación que toman en cuenta características estadísticas para definir los grupos, como por ejemplo al **codificación por desviación, de Helmert**, etc.

2.2 Manejo de valores faltantes

Muchas veces es posible que en un conjunto de registros como por ejemplo los ejemplificados en la siguiente tabla:

Para lidiar con los registros o tuplas faltantes, existen distintas técnicas:

1. **Ignorar las tuplas incompletas:** En circunstancias donde el porcentaje de tuplas incompletas es alto, es poco recomendable desecharlas. Los atributos incompletos pudieron ser de valor..
2. **Llenar los valores faltantes con constantes:** Por ejemplo, en atributos numéricos asignar la constante *Inf*. El uso de tales constantes puede confundir a muchos algoritmos de aprendizaje automático, cuando se calculan distancias, histogramas, etc.

3. **Usar la media, mediana o moda para rellenar tales valores:** Para atributos con valores de distribuciones no *torcidas* como la Gaussiana, se recomienda usar la media, mientras para datos con distribuciones *torcidas* como la distribución gamma, se recomienda usar la moda. El problema de utilizar el primer momento estadístico es la alteración de la desviación estándar de los datos.
 - (a) Una variante de este enfoque consiste en realizar un *clustering* de los datos, y asignar la media, moda o mediana del clúster correspondiente a cada registro con valor faltante.
4. **Usar el valor más probable:** realizando inferencia bayesiana o regresión, se puede estimar el valor más probable dados los valores del resto del registro, y también, del resto de los registros.
5. **Algoritmos más complejos como MICE (Multivariate imputation by chained equations),** el cual se basa en la idea de generar múltiples muestras de una regresión múltiple secuencial, tomando en cuenta que una muestra es generada de un fenómeno aleatorio, por lo que es una mejor aproximación generar múltiples muestras.

2.3 Transformaciones

Muchas veces, los datos originales pueden necesitar de funciones complejas para realizar ya sea una regresión o clasificación de los mismos, o incluso encajar alguna distribución de probabilidad. Por ejemplo, en la Figura 1, se muestra un conjunto de datos, para el cual es necesario realizar una regresión. Es fácil notar que un modelo lineal no logrará un error bajo. Sin embargo, dado que los datos parecen tener un comportamiento logarítmico, al aplicar la inversa de tal función a los datos y realizar la regresión en tales datos transformados, el modelo lineal es suficiente para lograr una regresión de error bajo. Tal función exponencial inversa no es más que una transformación

$$T(X)$$

En general, las técnicas de preprocesamiento a estudiar para señales e imágenes, tienen el objetivo de facilitar la extracción de características, clasificación o regresión, tal cual se muestra en la Figura 1.

2.4 Balanceo de datos

Un conjunto de datos desbalanceado, se refiere a un conjunto de datos $X \in \mathbb{R}^{n \times m}$, para el cual existen muchas más muestras de una clase respecto a las demás. Por ejemplo, en un problema de clasificación binaria, si las muestras de la clase C_1 se almacenan en la matriz $X \in \mathbb{R}^{n \times m_1}$, y las muestras de la clase C_2 están almacenadas en $X \in \mathbb{R}^{n \times m_2}$, se dice que el conjunto de datos está desbalanceado si $m_1 \gg m_2$ (o $m_2 \gg m_1$). Al realizar la implementación de

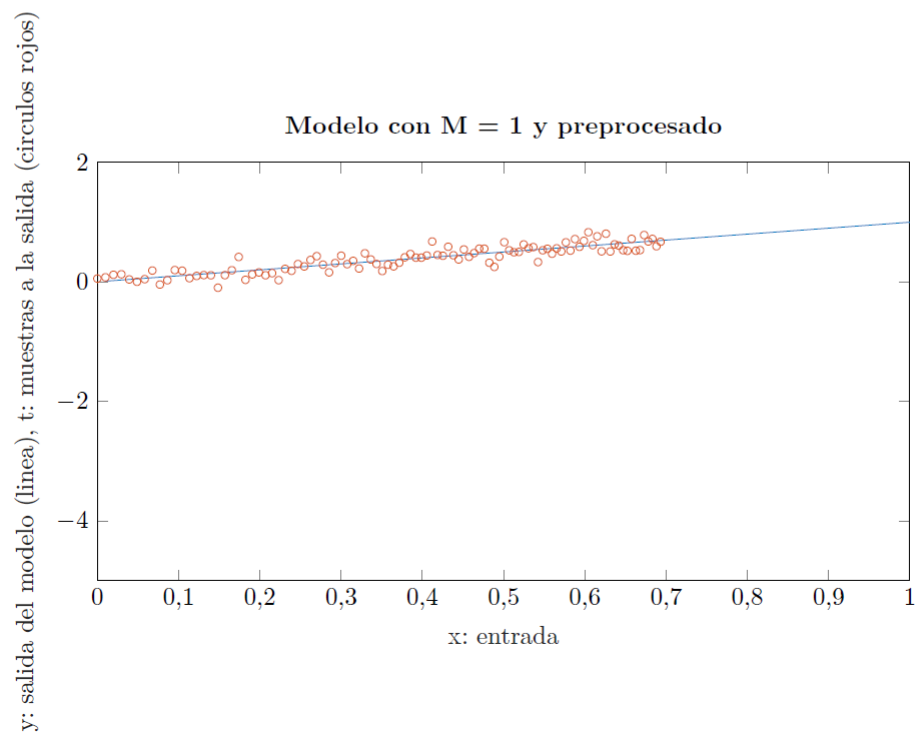
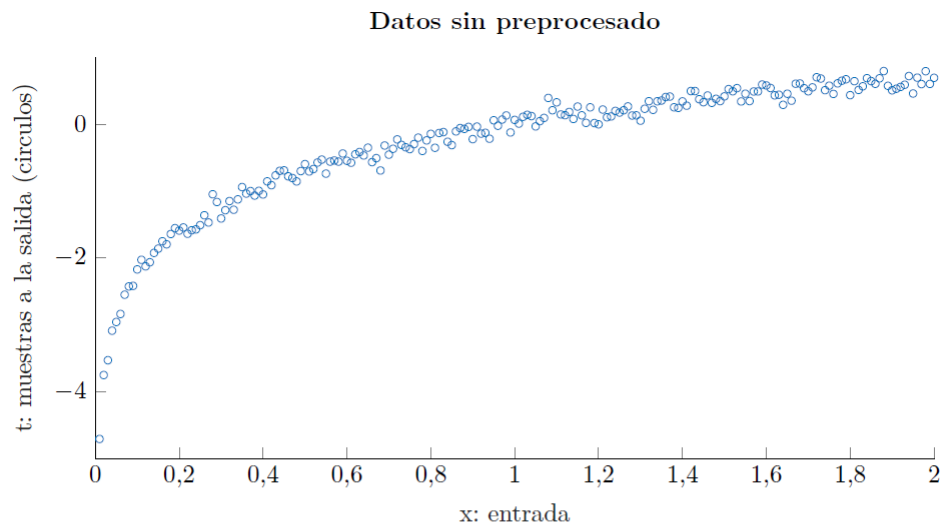


Figura 1: Regresión sin y con datos preprocesados.

un algoritmo de clasificación como mínimos cuadrados, los errores de ambas clases serían igualmente penados, por lo que es probable que el clasificador al que se arrije esté sesgado hacia a clase con mayor cantidad de muestras. Para corregir tal problema existen distintas técnicas posibles a implementar, manipulando el conjunto de datos X (suponga que $m_1 \gg m_2$):

1. **Sobre-muestreo u oversampling:** Consiste en generar las muestras faltantes para el conjunto de datos, repitiendo las muestras de la clase con menos muestras (C_2 en este caso), tantas veces sea necesario y de forma aleatoria.
2. **Sub-muestreo o subsampling:** Elimina las muestras sobrantes de la clase con mayor número de muestras, en este caso m_1 .
3. **Generación de nuevas muestras usando aprendizaje generativo:** El **aprendizaje generativo** se enfoca en aprender la distribución de los datos, a diferencia del aprendizaje discriminativo, el cual tiene por objetivo construir un modelo que disminuya el error de clasificación o discriminación.

Otro enfoque, no basado en la generación o eliminación de nuevas muestras, consiste en modificar la función de pérdida del modelo para *castigar* con mayor fuerza los errores con muestras de las clases con menos muestras. Sin embargo tal enfoque no corresponde al de una etapa de preprocesamiento.

3 Preprocesamiento de señales, series temporales e imágenes

Una señal puede definirse como una serie de valores, con origen usualmente de fenómenos físicos: sonidos, correspondientes a alteraciones en la presión del aire, intensidades lumínicas en distintas longitudes de onda, de origen biológico como la presión arterial, etc. Existen tres grandes grupos, que se resumen en el siguiente cuadro:

Señales	Espacio	Amplitud	Ejemplos
Analógicas	Continua	Continua	Señales sísmicas, electrocardiogramas, ondas cerebrales, sonido, etc.
Discretas	Discreta	Continua	Señales muestreadas en dispositivos analógicos
Digitales	Discreta	Discreta	Imágenes, señales y sonidos en un computador digital

Cuadro 3: Tipos de señales.

Las señales naturalmente son analógicas, y dado que gran parte de los computadores actuales son digitales, es necesario realizar el proceso de muestreo, el cual consiste en dos etapas: discretizar primero el dominio del tiempo,

tomando una cantidad de muestras finitas y almacenandolas en un dispositivo analógico como un condensador, para posteriormente realizar la etapa de cuantificación (codificación en binario) de las muestras, de modo que la amplitud sea también discreta. La Figura 2 muestra el diagrama de un circuito de muestreo en el tiempo y en amplitud básico.

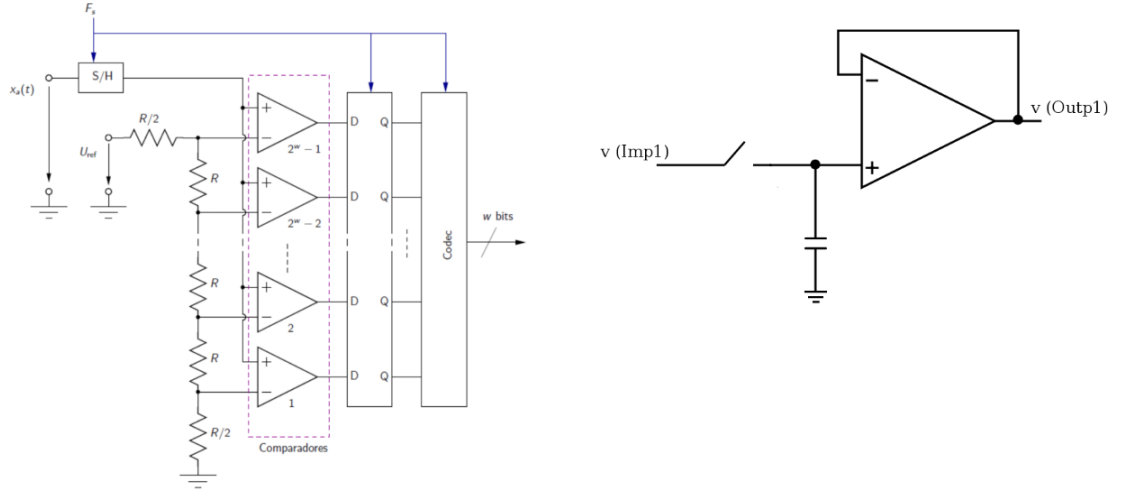


Figura 2: Circuito de muestreo.

La Figura 3 muestra el proceso de digitalización de una señal mencionado. En el primer gráfico, únicamente el dominio del tiempo es discreto, en el segundo, tanto el dominio del tiempo como el de la intensidad o amplitud de la señal son discretos, lo que corresponde entonces a una señal digital.

3.1 Seno y coseno

La función sinusoidal o seno es una función impar con período 2π (π corresponde a la razón entre la circunferencia y el diámetro de un círculo) la cual oscila suavemente hacia el infinito, y está dada en su forma más sencilla como:

$$f(x) = \sin(x),$$

y su gráfica se muestra en la Figura 4. Observe que la función se encuentra entre 1 y -1 , lo que se refiere a la **amplitud del seno**, y efectivamente tiene un período de $2\pi \cong 6,2832$. Otras formas frecuentes de la función sinusoidal son $f_1(x) = \sin(\pi x)$ y $f_2(x) = \sin(2\pi x)$, las cuales se grafican en la Figura 5. Como se puede observar, el aumentar el coeficiente de la variable libre x acorta el período de repetición de la función sinusoidal T , el cual para la función f_1 corresponde a $T = 2$ y para la función f_2 a $T = 1$. Por esta razón es usual

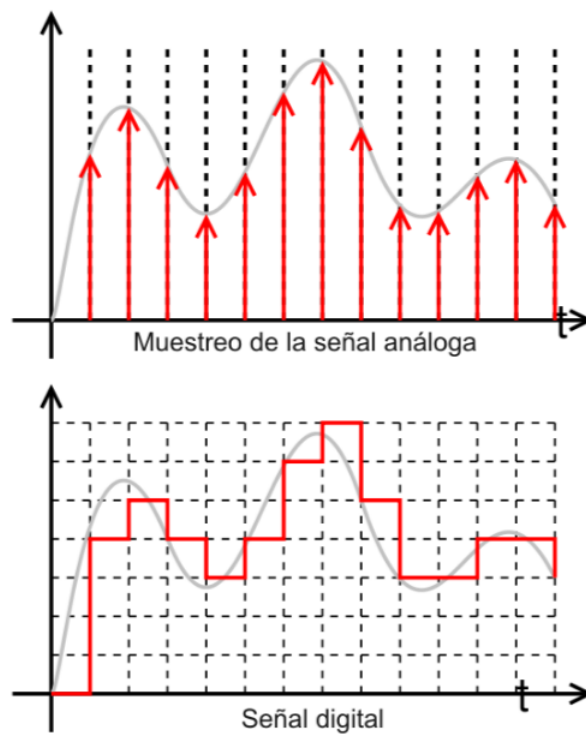


Figura 3: Señal digital. Tomado de <https://ccvredes1.wordpress.com/2012/10/06/senales-y-codificacion-parte-iv/>

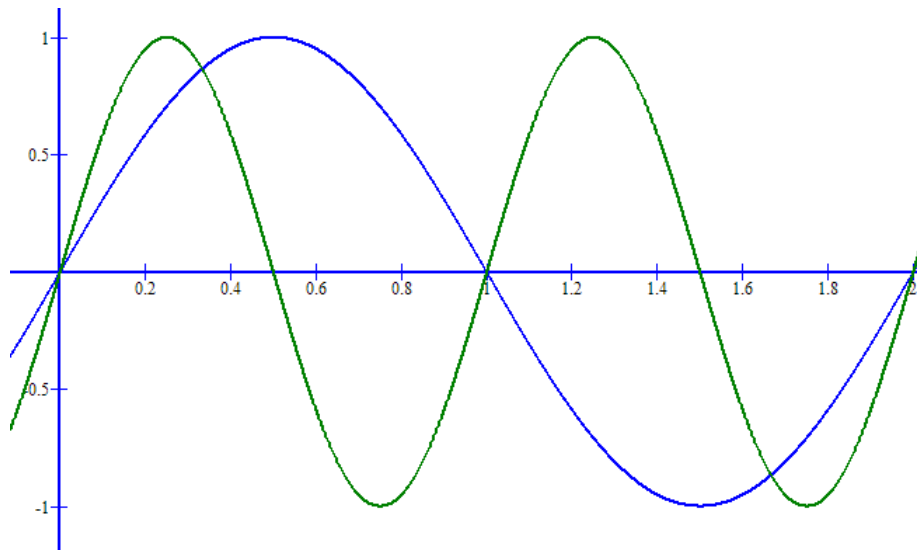


Figura 5: Funciones $f(x) = \sin(\pi x)$ (azul) y $f(x) = \sin(2\pi x)$ (verde).

encontrar la forma de f_2 para la función seno, puesto que tanto su amplitud $a = 1$ y su período es $T = 1$.

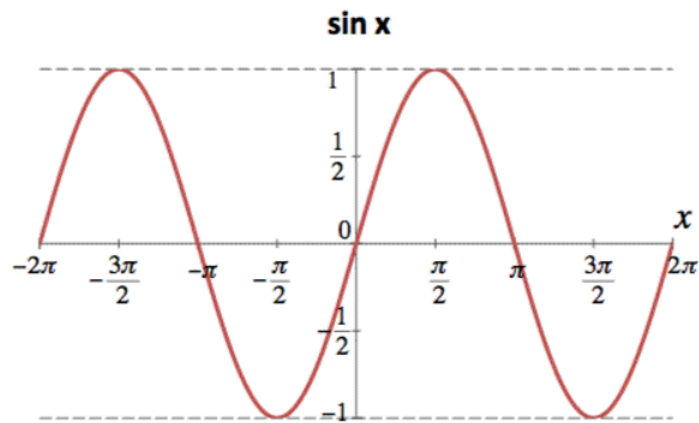


Figura 4: Función seno.

La forma general de la función seno viene dada por:

$$f(t) = a \sin(\omega f t + \phi),$$

con $\omega = 2\pi$ donde f define la frecuencia de oscilación de la función seno. Suponiendo que el codominio de la función está dado en segundos (s) observe

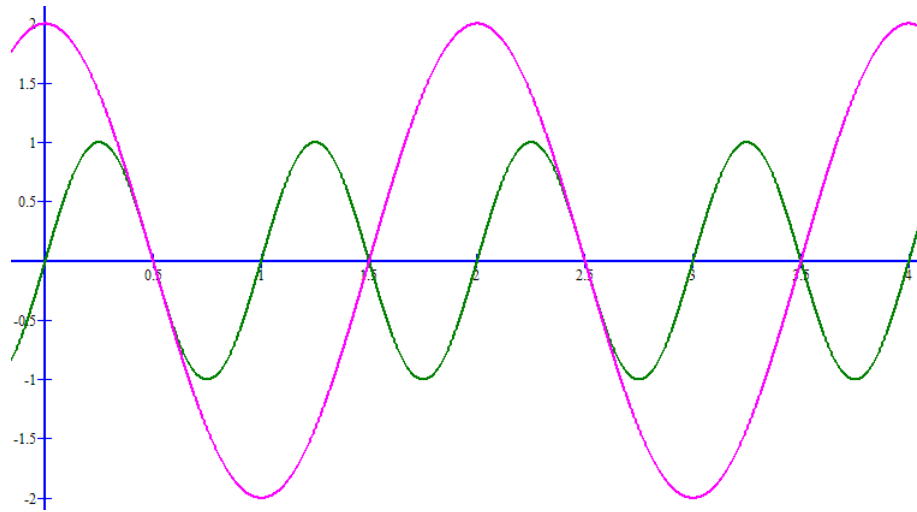


Figura 6: Graficación de las funciones f_2 (verde) y f_3 (magenta) con distinta amplitud, fase y frecuencia.

a la función $f_3(t) = 2 \sin(\omega \frac{1}{2} t + \frac{\pi}{2})$, la cual presenta un período $T = 2s$, una amplitud $a = 2$ y un cambio de fase o desplazamiento $\phi = \frac{\pi}{2}$, el cual en la función equivale a un retraso de la onda por $\frac{\phi}{\omega f}$ (lo que en este caso corresponde a un desplazamiento por $-\frac{1}{2}$), como se observa en la Figura 6. Se deduce además la relación inversa entre el período y la frecuencia:

$$f = \frac{1}{T}$$

En general, el **coseno** (función par) corresponde a una función sinusoidal con un desplazamiento por $-\frac{1}{2}$, por lo que entonces:

$$\sin\left(2\pi x + \frac{\pi}{2}\right) = \cos(2\pi x)$$

Las derivadas e integrales del seno y coseno están dadas como sigue:

$$\begin{aligned} \frac{d}{dx} \sin(\pi x) &= \pi \cos(\pi x) & \int \sin(\pi x) dx &= -\frac{1}{\pi} \cos(\pi x) \\ \frac{d}{dx} \cos(\pi x) &= -\pi \sin(\pi x) & \int \cos(\pi x) dx &= \frac{1}{\pi} \sin(\pi x) \end{aligned}$$

Para el caso de la integral definida

$$\int_{-\infty}^{\infty} \sin(\pi x) dx$$

formalmente diverge. Sin embargo, si consideramos que físicamente la onda sinusoidal es finita (matemáticamente podemos pensar que la función puede acotarse por alguna función par $P(x)$ de modo que $g(x) = P(x)$), y examinando que en un período la integral de la función es negativa y positiva en igual cantidad de intervalos, se puede considerar que:

$$\int_{-\infty}^{\infty} \sin(\pi x) dx \cong \int_{-\infty}^{\infty} \cos(\pi x) dx \cong 0,$$

un resultado útil como se verá más adelante.

3.2 Función impulso

La función impulso o también conocida como función *delta dirac* o *delta Kronecker* $\delta(x)$ no es estrictamente una función matemática, sino más bien una abstracción, que fija las siguientes propiedades para la función:

$$\delta(x) = \begin{cases} 0 & \text{si } x \neq 0 \\ \infty & x = 0 \end{cases}$$

y

$$\int_{-\infty}^{\infty} \delta(x) dx = 1.$$

La representación gráfica corresponde a una flecha en $x = 0$ representando el hecho de que en el eje y la función tiende al infinito, y el eje x la función tiende a cero, pero a pesar de ello, manteniendo un área bajo la curva de 1.

La necesidad de la abstracción de la función impulso deriva con lo usual de encontrar magnitudes físicas que concentran mucha energía en un instante de tiempo muy corto o espacio muy pequeño. La función impulso se puede aproximar a partir por ejemplo, de una función rectángulo (la cual se trata en la siguiente subsección):

$$\int_{-\infty}^{\infty} \delta(x) dx \cong \lim_{\tau \rightarrow 0} \frac{1}{\tau} \int_{-\infty}^{\infty} \Pi\left(\frac{x}{\tau}\right) dx.$$

Observe que el multiplicar la función rectangular por $\frac{1}{\tau}$ hace que la función rectángulo, originalmente con amplitud unitaria, tenga una altura de $\frac{1}{\tau}$ y el dividir el argumento por τ hace que la base del rectángulo sea de magnitud τ . La Figura 10 muestra como a partir de la función rectangular se aproxima al impulso, conforme $\tau \rightarrow \infty$. En realidad, la función impulso se puede aproximar con cualquier función simétrica con valor mayor que cero alrededor del origen, y decreciente conforme se aleja, como por ejemplo una función Gaussiana, ilustrado en la animación de la Figura 8.

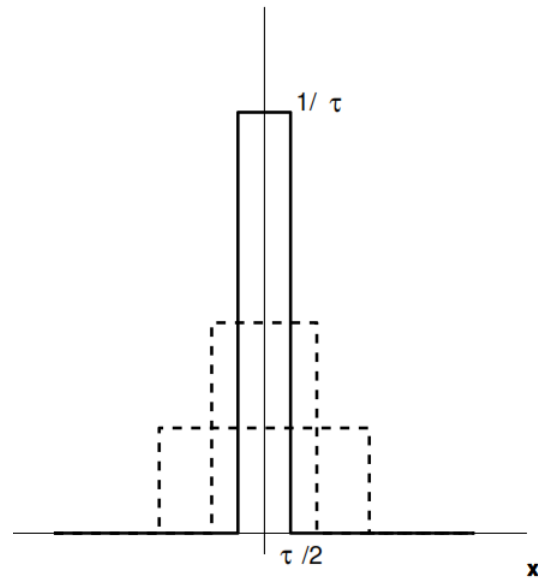


Figura 7: Aproximación con la función *rect* del impulso. Tomado de [2]

(Loading...)

Figura 8: Aproximación de la función impulso con una función Gaussiana.

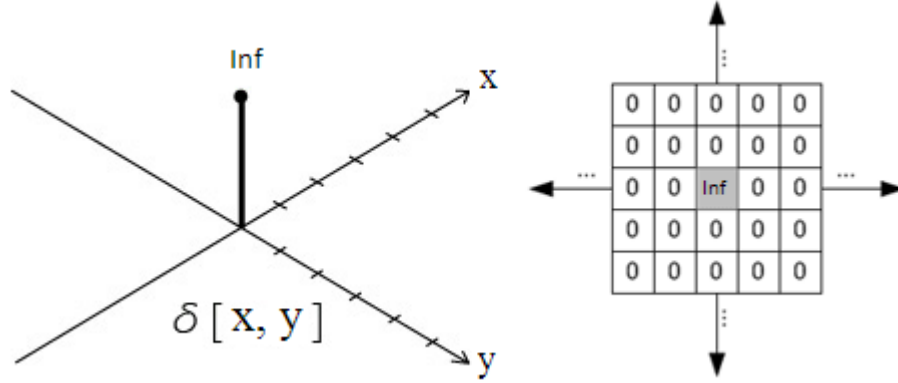


Figura 9: Graficación de la función impulso.

3.2.1 Función impulso con dominio de dos dimensiones

La función impulso $\delta(x, y)$ definida en un dominio en \mathbb{R}^2 se define como:

$$\delta(x, y) = \begin{cases} 0 & \text{si } x \neq 0 \text{ o } y \neq 0 \\ \infty & x = y = 0 \end{cases}$$

y

$$\int_{-\infty}^{\infty} \delta(x, y) dx dy = 1.$$

y puede graficarse como se muestra en la Figura 9.

3.2.2 La propiedad del cedazo

Observe la siguiente integral:

$$\int_{-\infty}^{\infty} \delta(x) f(x) dx,$$

la cual, reemplazando la función *delta dirac* por su equivalente en términos de la función *rect* vendría dada por:

$$\frac{1}{\tau} \int_{-\infty}^{\infty} \Pi\left(\frac{x}{\tau}\right) f(x) dx = \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \frac{1}{\tau} f(x) dx,$$

y recordemos que $\tau \rightarrow 0$, por lo que entonces la integral es distinta de cero en el intervalo $-\frac{\tau}{2}$ a $\frac{\tau}{2}$ (según la definición de la función *rect*), con lo que entonces se tiene que:

$$\int_{-\infty}^{\infty} \delta(x) f(x) dx = f(0)$$

y en general, para un impulso centrado en a :

$$\int_{-\infty}^{\infty} \delta(x-a) f(x) dx = f(a)$$

y cuando la función f está desplazada por a :

$$\int_{-\infty}^{\infty} \delta(x) f(x-a) dx = f(-a).$$

A esta propiedad se le llama la propiedad del cedazo, pues extrae un valor puntual de la función f mediante la función impulso.

3.3 Función rectángulo

La función conocida como rectangular o rect, la cual se muestra en la Figura 10 :

$$\square(x) = \begin{cases} 1 & \text{si } |x| \leq \frac{1}{2} \\ 0 & \text{sino} \end{cases}$$

Ello significa entonces que

$$\int_{-\infty}^{\infty} \square(x) dx = 1$$

En los puntos $x = -\frac{1}{2}$ y $x = \frac{1}{2}$ se puede afirmar que existe una discontinuidad, por lo que la derivada no está definida. Sin embargo, se puede suponer una transición más suave en tales puntos, como la línea punteada dibujada en la Figura 10, función que etiquetaremos como $\square_{\alpha}(x)$, de modo que la derivada en la práctica tienda a ∞ . La constante α define la *suavidad* para la aproximación de la función rectangular.

De este modo la derivada de la función *rect* en los puntos $x = -\frac{1}{2}$ y $x = \frac{1}{2}$ se puede aproximar como:

$$\frac{d}{dx} \square(x) = \delta\left(x + \frac{1}{2}\right) - \delta\left(x - \frac{1}{2}\right).$$

Una modificación usual que conserva el valor de la integral unitario es la siguiente:

$$\frac{1}{\tau} \square\left(\frac{x}{\tau}\right) = \begin{cases} \frac{1}{\tau} & \text{si } |x| \leq \frac{\tau}{2} \\ 0 & \text{sino} \end{cases},$$

de esta forma, como se vió en la Figura 7, conforme $\tau \rightarrow 0$, se tiene que $\frac{1}{\tau} \square\left(\frac{x}{\tau}\right) \rightarrow \delta(x)$.

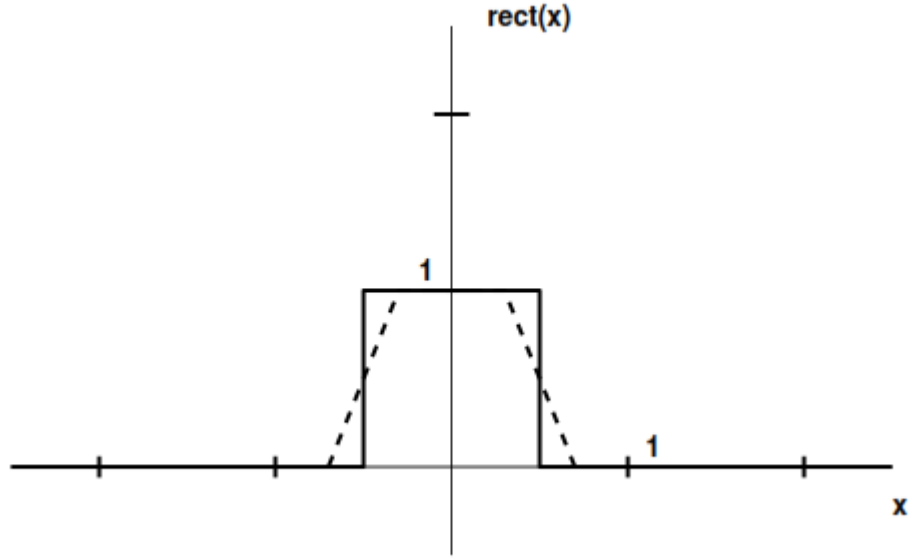


Figura 10: Función $\text{rect} \sqcap(x)$ y su aproximación $\sqcap_{\alpha}(x)$ en línea punteada. Tomado de [2]

3.3.1 Función rectángulo con dominio de dos dimensiones

La función $\text{rect} \sqcap(x, y)$ en dos dimensiones se define como:

$$\sqcap(x, y) = \begin{cases} 1 & \text{si } |x| \leq \frac{1}{2} \text{ y } |y| \leq \frac{1}{2} \\ 0 & \text{sino} \end{cases}.$$

La Figura 11 muestra la graficación de la función rect desde dos perspectivas (lateral y desde arriba, en una imagen).

3.4 Función escalón o de Heaviside

La función escalón o también conocida como de Heaviside $\ulcorner(x)$ se define como:

$$\ulcorner(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Recordando la definición de la derivada para la función rect , para el caso de la función de Heaviside, se define su derivada como:

$$\frac{d}{dx} \ulcorner(x) = \delta(x)$$

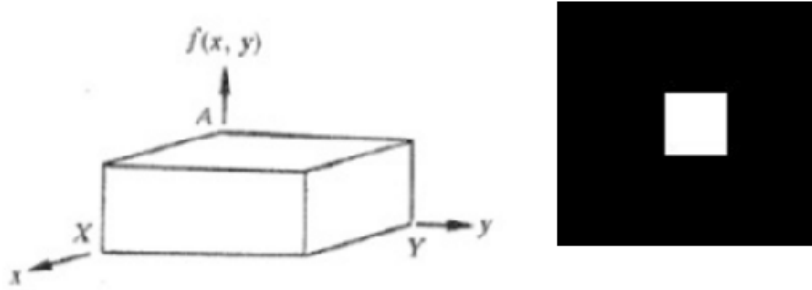


Figura 11: Función *rect* con dominio en dos dimensiones.

Para hacer que la función esté centrada en a , como es usual, se hace $\cap(x - a)$, con lo que:

$$\cap(x) = \begin{cases} 1 & \text{si } x > a \\ 0 & \text{si } x < a \end{cases}$$

3.5 Función signo

La función signo, representada como $\text{sgn}(x)$, está definida como sigue:

$$\text{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x < 0 \end{cases}$$

Para la cual la derivada está dada por:

$$\frac{d}{dx} \text{sgn}(x) = \delta(x)$$

Y para hacer que la función esté centrada en a , como es usual, se hace $\cap(x - a)$, con lo que:

$$\cap(x - a) = \begin{cases} 1 & \text{si } x > a \\ 0 & \text{si } x < a \end{cases}$$

La Figura 12 muestra las gráficas de las funciones signo y Heaviside.

3.6 Función sinc o seno cardenal

La función *sinc* o conocida también como seno cardenal se define como sigue:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x},$$

y cruza por cero cuando $|x| = 1, 2, 3, \dots$, su valor cuando $x = 0$ es uno (lo cual se demuestra más abajo, por L'Hopital), y el área total bajo la curva es uno también:

$$\text{sinc}(0) = \lim_{x \rightarrow 0} \frac{\sin(\pi x)}{\pi x} = \lim_{x \rightarrow 0} \frac{\cos(\pi x)}{\pi x} = 1,$$

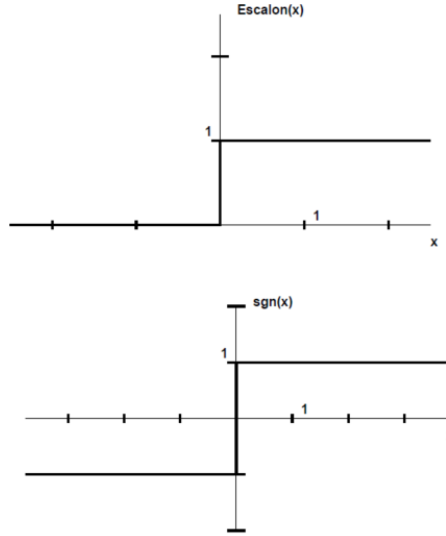


Figura 12: Funciones signo y de Heaviside. Tomado de [2]

$$\int_{-\infty}^{\infty} \text{sinc}(x) dx = 1.$$

Todo ello se ilustra en la Figura 13. Las ondas con cruces por cero se les llama lóbulos, y tienen magnitudes decrecientes (el lóbulo central tiene magnitud 1, el segundo 0, 21, y el tercero 0, 13 aproximadamente).

4 Convolución

La convolución es una operación muy común en el procesamiento de señales. Implementa el concepto de “ventana deslizante” en el que básicamente una función se desliza sobre otra, y se calcula la respuesta de multiplicarlas en cada punto del “deslizamiento”. Formalmente, la integral de convolución $h(x)$ de dos funciones $f(x)$ y $g(x)$ se define como:

$$h(x) = f(x) * g(x) = \{f * g\}(x) = \int_{-\infty}^{\infty} f(\tau) g(x - \tau) d\tau$$

Observe que el término $g(x - \tau)$ corresponde a una traslación de la función $g(x)$ por τ unidades, con τ asociado a la variable de integración que va de $-\infty$ a ∞ .

Como ejemplo, considere la función $\text{rect} \sqcap(x)$, convolucionada consigo misma:

$$h(x) = \sqcap(x) * \sqcap(x) = \int_{-\infty}^{\infty} \sqcap(\tau) \sqcap(x - \tau) d\tau.$$

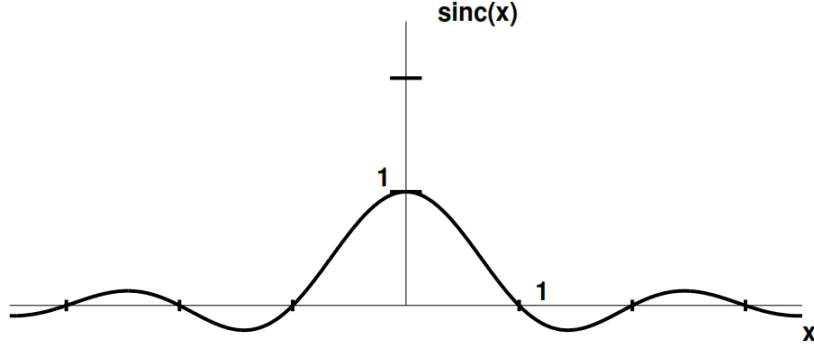


Figura 13: Función de seno cardinal.

Observemos el proceso de convolución ilustrado gráficamente en la Figura 14. Tome el ejemplo de evaluar la convolución en $x = -500$ (un valor muy negativo, tendiendo a $-\infty$)

$$h(-500) = \int_{-\infty}^{\infty} \square(\tau) \square(-500 - \tau) d\tau = \int_{-\infty}^{\infty} \square(\tau) \square(-(500 + \tau)) d\tau,$$

con la equivalencia anterior es fácil ver que la función $\square(-(500 + \tau))$ tiene su origen en $\tau = -500$, por lo que está centrada en tal punto, y además, como su argumento está negado, la función está reflejada respecto a su origen, sin embargo al ser una función simétrica, ello no tiene ningún efecto.

La integral de menos infinito a infinito, suma la multiplicación de todos los puntos, con lo que en este caso, dado que no existe intersección entre ambas funciones, la integral resulta en cero. Lo anterior se ilustra en la primer sección de arriba a abajo de la Figura 14. Posteriormente, conforme x se acerca a cero, la integral de la intersección entre ambos rectángulos comienza a aumentar, lo que hace que la función $h(x)$ comience a crecer y alcanza su máximo cuando $h(0)$ (correspondiente a $x = 0$), pues en tal caso se tiene la siguiente integral:

$$h(0) = \int_{-\infty}^{\infty} \square(\tau) \square(-\tau) d\tau = \int_{-\infty}^{\infty} \square(\tau) \square(\tau) d\tau = \int_{-\frac{1}{2}}^{\frac{1}{2}} 1 d\tau = 1,$$

lo cual implica que la función de convolución tiene su máximo en $x = 0$, puesto que conforme $x \rightarrow \infty$, la integral decrece hasta ser nula cuando no existe intersección entre ambos rectángulos, lo cual lo ilustra también la Figura 14.

La Figura 15 muestra una animación del proceso de convolucionar dos rectángulos pero de distinta base, lo que resulta en un trapecio.

Otro ejemplo de la convolución es la convolución de una función cualquiera f con la función impulso:

$$h(x) = f(x) * \delta(x) = \int_{-\infty}^{\infty} f(\tau) \delta(x - \tau) d\tau.$$

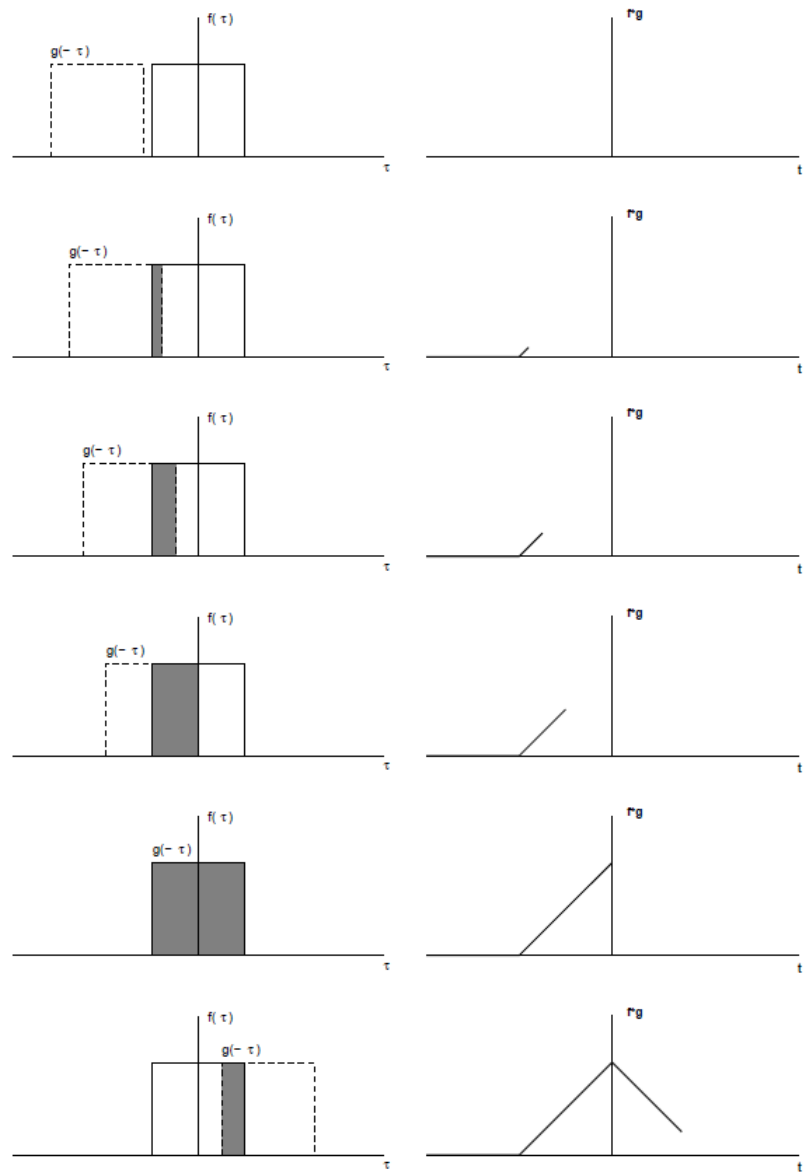


Figura 14: Distintos puntos de evaluación para la convolución. Tomado de [2]

(Loading...)

Figura 15: Convolución de dos rectángulos de distinta base.

Recuerde que para un punto cualquiera $x = a$, se tiene que $f(\tau) \delta(a - \tau) = f(a)$ por la propiedad del cedazo, por lo que entonces, la convolución anterior corresponde a aplicar la propiedad del cedazo para todos los puntos en x , por lo que se tiene que:

$$h(x) = f(x) * \delta(x) = f(x).$$

Observe además que tomando la convolución:

$$h(x) = \delta(x) * f(x) = \int_{-\infty}^{\infty} \delta(\tau) f(x - \tau) d\tau = f(x),$$

lo cual sugiere la conmutatividad de la convolución.

La convolución discreta para dos funciones discretas $f[n]$ y $g[n]$ se define como:

$$h[n] = f[n] * g[n] = \sum_{m=-\infty}^{\infty} f[m] g[n - m]$$

Ejemplo 1

Sean las funciones $f[n] = (10 \ 10 \ 10)$ y $g[n] = (\frac{1}{2} \ \frac{1}{2})$, su convolución viene dada entonces por:

$$h[n] = (10 \ 10 \ 10) * (\frac{1}{2} \ \frac{1}{2}).$$

Tomemos en cuenta que $h[n]$ se encuentra definida para $n \geq 0$ y lo mismo sucede para las funciones $f[n]$ y $g[n]$. Si no lo está, se asume que su valor es **cero**. Esto implica que, por ejemplo, para obtener el valor de $h[0]$ (usando a cero como el índice inicial) se tiene que:

$$h[0] = \sum_{m=0}^0 f[m] g[0 - m] = f[0] \cdot g[0] = 10 \cdot \frac{1}{2} = 5$$

y para otros casos, de forma similar se tiene que:

$$\begin{aligned} h[1] &= \sum_{m=0}^1 f[m] g[1-m] = f[0] \cdot g[1] + f[1] \cdot g[0] = 10 \cdot \frac{1}{2} + 10 \cdot \frac{1}{2} = 10 \\ h[2] &= \sum_{m=0}^2 f[m] g[2-m] = f[1] \cdot g[1] + f[2] \cdot g[0] = 10 \cdot \frac{1}{2} + 10 \cdot \frac{1}{2} = 10 \\ h[3] &= \sum_{m=0}^3 f[m] g[3-m] = f[2] \cdot g[1] = 10 \cdot \frac{1}{2} = 5 \end{aligned}$$

Lo cual si se grafica, tiene forma similar a lo mostrado en la Figura 14.

Ejemplo 2

Sean las funciones $f[n] = (1 \ 2 \ 4 \ 3 \ 2)$ y $g[n] = (1 \ 2 \ 2)$, su convolución viene dada entonces por:

$$h[n] = (1 \ 2 \ 4 \ 3 \ 2) * (1 \ 2 \ 2) = (1 \ 4 \ 10 \ 15 \ 16 \ 10 \ 4).$$

Para entender como se obtuvo tal resultado, tomemos en cuenta que $h[n]$ se encuentra definida para $n \geq 0$ y lo mismo sucede para las funciones $f[n]$ y $g[n]$. Esto implica que, por ejemplo, para obtener el valor de $h[0]$ (usando a cero como el índice inicial) se tiene que:

$$h[0] = \sum_{m=0}^0 f[m] g[0-m] = f[0] \cdot g[0] = 1 \cdot 1 = 1$$

y para otros casos, de forma similar se tiene que:

$$\begin{aligned} h[1] &= \sum_{m=0}^1 f[m] g[1-m] = f[0] \cdot g[1] + f[1] \cdot g[0] = 1 \cdot 2 + 2 \cdot 1 = 4 \\ h[2] &= \sum_{m=0}^2 f[m] g[2-m] = f[0] \cdot g[2] + f[1] \cdot g[1] + f[2] \cdot g[0] = 1 \cdot 2 + 2 \cdot 2 + 4 \cdot 1 = 10 \end{aligned}$$

4.1 Correlación cruzada

La correlación cruzada es una **medida de similitud** entre las funciones reales f y g y viene dada por:

$$c(x) = f(x) \star g(x) = \{f \star g\}(x) = \int_{-\infty}^{\infty} f(\tau) g(x + \tau) d\tau$$

y en su forma discreta viene dada por:

$$c[n] = f[n] \star g[n] = \sum_{m=-\infty}^{\infty} f[m] g[n + m]$$

y equivale a realizar la convolución sin reflejar o rotar la función g . Observe que si la función g es simétrica la correlación y la convolución son equivalentes:

$$f(x) \star g(x) = f(x) * g(x)$$

4.2 Autocorrelación

La autocorrelación de una función f equivale a la correlación de la función f consigo misma:

$$c(x) = f(x) \star f(x) = \int_{-\infty}^{\infty} f(\tau) f(x + \tau) d\tau$$

y en forma discreta:

$$c[n] = f[n] \star f[n] = \sum_{m=-\infty}^{\infty} f[m] f[n + m]$$

la autocorrelación es utilizada en el procesamiento de señales para averiguar si una función f es periódica o no, pues si c tiene dos o más máximos de igual magnitud, la señal fue equivalente dos o más veces a sí misma, con un desplazamiento.

4.3 Convolución con dominio \mathbb{R}^2

La convolución en dos ejes se define para dos funciones continuas $g(x, y)$ y $f(x, y)$ como:

$$(g * f)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x - u, y - v) f(u, v) du dv$$

y para dos funciones discretas $g[x, y]$ y $f[x, y]$ como:

$$(g * f)[x, y] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} g[x - u, y - v] f[u, v] \quad (1)$$

En el presente curso, será usual utilizar matrices digitales (muestras finitas de valores discretos), por lo que los funcionales con dominio definido en \mathbb{R}^2 y codominio en \mathbb{R} serán representados como tales matrices digitales: $U \in \mathbb{R}^{N \times M}$ y $F \in \mathbb{R}^{K \times L}$, con $F, G : \mathbb{R}^2 \rightarrow \mathbb{R}$ y respecto a las dimensiones $K < N$ y $L < M$. Para usar matrices con origen en 0,0, se utiliza la siguiente definición y una notación que prescinda de rotaciones del núcleo F (lo cuál lo hace similar a la correlación cruzada):

$$(U * F)[x, y] = \sum_{u=0}^{K-1} \sum_{v=0}^{L-1} U \left[x - \frac{(K-1)}{2} + u, y - \frac{(L-1)}{2} + v \right] F[u, v]$$

Ejemplo 1:

Sean las matrices

$$U = \begin{bmatrix} 2 & 3 & 4 & 1 & 5 \\ 2 & 9 & 20 & 1 & 5 \\ 1 & 5 & 3 & 7 & 8 \\ 1 & 2 & 3 & 7 & 5 \\ 5 & 7 & 8 & 9 & 10 \end{bmatrix} \quad F = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

La respuesta de la convolución

$$(U * F)[2, 2] = \sum_{u=0}^2 \sum_{v=0}^2 U[2-1+u, y-1+v] F[u, v]$$

$$(U * F)[2, 2] = \begin{matrix} U[1, 1] F[0, 0] + U[1, 2] F[0, 1] + U[1, 3] F[0, 2] \\ U[2, 1] F[1, 0] + U[2, 2] F[1, 1] + U[2, 3] F[1, 2] \\ U[3, 1] F[2, 0] + U[3, 2] F[2, 1] + U[3, 3] F[2, 2] \end{matrix}$$

$$(U * F)[2, 2] = \begin{matrix} 9 \cdot -1 + 20 \cdot 0 + 1 \cdot 1 \\ 5 \cdot -1 + 3 \cdot 0 + 7 \cdot 1 \\ 2 \cdot -1 + 3 \cdot 0 + 7 \cdot 1 \end{matrix} = -1$$

El resultado es realizar el producto punto de la sub-matriz $U_{1:3,1:3}$ con el filtro F . Observe que por ejemplo, la respuesta de la convolución en $(U * F)[0, 0]$ no es posible de calcular, lo cual genera un efecto indeseable de cambio de la dimensionalidad en la matriz resultante de la convolución. Es por esto que es frecuente concatenar filas y columnas llenas de ceros (0's) en los bordes de la matriz con mayores dimensiones U , lo cual se refiere como *padding*. La cantidad de filas con ceros a concatenar se representa con Δf y la cantidad de columnas con Δc , por lo que entonces la concatenación de ceros en filas y columnas está dada por la operación P:

$$p(U) = U' \in \mathbb{R}^{(N+2z_f) \times (M+2z_c)}$$

, por lo que para preservar la dimensionalidad de la matriz resultante en la convolución, se hace que: $z_c = \frac{(K-1)}{2}$ y $z_f = \frac{(L-1)}{2}$.

La definición de la convolución según la Ecuación 3 se ilustra en la Figura 16, donde se observa como la función con menor dimensionalidad básicamente se *desliza* sobre la función con mayor dimensionalidad (para ello, por convención, se fija como el segundo operando). Por cada *deslizamiento* de la función G sobre F , se calcula el **producto punto entre ambas matrices**, centrado en el elemento actual (x, y) de tal deslizamiento:

$$(U * F)[x, y] = \alpha U_{x-\frac{(K-1)}{2} : x+\frac{(K-1)}{2}, y-\frac{(L-1)}{2} : y+\frac{(L-1)}{2}} \cdot F, \quad (2)$$

donde el **producto punto se puede definir para dos matrices** $A, B \in \mathbb{R}^{N \times M}$ como:

$$A \cdot B = \sum_{i=0}^N \sum_{j=0}^M A_{i,j} B_{i,j}$$

con $s \in \mathbb{R}$.

- **Ejemplo:** Sean las matrices o funciones F y $U, F, U : \mathbb{R}^2 \rightarrow \mathbb{R}$ dadas por:

$$U = \begin{bmatrix} 2 & 3 & 2 \\ 3 & 25 & 3 \\ 2 & 3 & 2 \end{bmatrix} \quad F = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

calcule la **convolución normalizada**. Realice el *padding* necesario para conservar la dimensionalidad de la señal. En este caso se tiene que $K = L = 3$, $\alpha = \frac{1}{9}$. A la señal U se le realiza una concatenación de una fila y columna de ceros, de forma que las columnas negativas en una unidad sean cero, lo que resulta en:

$$U' = p(U) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 2 & 0 \\ 0 & 3 & 25 & 3 & 0 \\ 0 & 2 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

por lo que entonces se tiene que:

$$\begin{aligned} (U * F)[0, 0] &= (U' * F)[1, 1] = \frac{1}{9} \sum_{u=0}^2 \sum_{v=0}^2 U'[1-u, 1-v] F[u, v] = U'_{0:2, 0:2} \cdot F \\ &= \frac{1}{9} U'[0, 0] F[0, 0] + U'[0, 1] F[0, 1] + U'[0, 2] F[0, 2] \\ &\quad + U'[1, 0] F[1, 0] + U'[1, 1] F[1, 1] + U'[1, 2] F[1, 2] \\ &\quad + U'[2, 0] F[2, 0] + U'[2, 1] F[2, 1] + U'[2, 2] F[2, 2] = 3,6667 \end{aligned}$$

La notación de producto punto es más clara, veamos como se calcula la respuesta en $[0, 1]$:

$$(U * F)[0, 1] = \alpha U_{0-1:0+1, 1-1:1+1} \cdot F = \frac{1}{9} U_{-1:1, 0:2} \cdot F = \frac{1}{9} \begin{bmatrix} 0 & 0 & 0 \\ 2 & 3 & 2 \\ 3 & 25 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = 4,2222$$

Realizando todos los cálculos para todas las entradas, se tiene que:

$$(U * F) = \begin{bmatrix} 3,66 & 4,22 & 3,66 \\ 4,22 & 5 & 4,22 \\ 3,66 & 4,22 & 3,66 \end{bmatrix}$$

En MATLAB se puede usar la función *conv2*, la cual, por defecto, hace el padding para las dos señales, conservando la dimensionalidad a la salida.

```
F = [1/9 1/9 1/9; 1/9 1/9 1/9; 1/9 1/9 1/9];
U = [2 3 2; 3 25 3; 2 3 2]
C = conv2(U, F);
```

Muchas veces, con el afán de ahorrar recursos computacionales, el deslizamiento de la función U sobre F se hace con un *paso* mayor (en inglés conocido como *stride*), de forma que el deslizamiento se hace a *saltos*. La cantidad de filas y columnas que se salta en el deslizamiento se define por Δ , la convolución se puede reescribir como:

$$(U * F)[x, y] = \sum_{u=0}^{K-1} \sum_{v=0}^{L-1} U \left[x \Delta - u + \frac{(K-1)}{2}, y \Delta - v + \frac{(L-1)}{2} \right] F[u, v]. \quad (3)$$

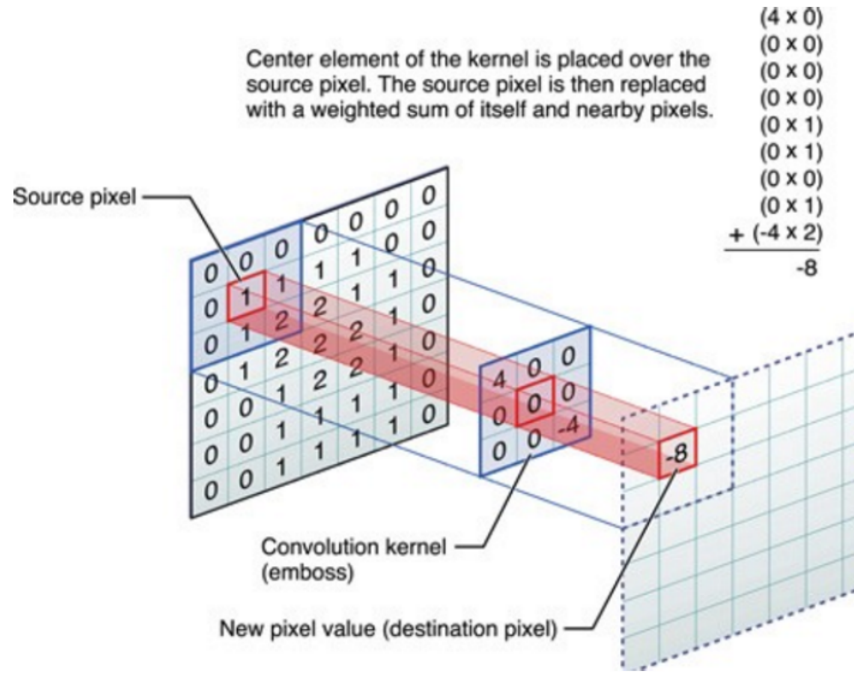


Figura 16: Deslizamiento en la convolución. Tomado de *Deep evolution of image representations for handwritten digit recognition*

lo cual reduce la dimensionalidad de la matriz resultante de la convolución, submuestreando el resultado original.

En las páginas https://github.com/vdumoulin/conv_arithmetic/blob/master/README.md y <http://cs231n.github.io/convolutional-networks/> se muestra la operación de la convolución animada.

En general, para determinar la dimensionalidad de la imagen convolucionada o la salida $Y \in \mathbb{R}^{A \times B}$, con una matriz de entrada $U \in \mathbb{R}^{N \times M}$ y un filtro $F \in \mathbb{R}^{K \times L}$, con un paso Δ y un *padding* z , se puede utilizar la siguiente formulación:

$$A = \frac{N - K + 2z}{\Delta} + 1, \quad B = \frac{M - L + 2z}{\Delta} + 1$$

4.4 Propiedades de la convolución

Las siguientes son propiedades generales de la convolución y convoluciones conocidas:

- Conmutatividad $f(x) * g(x) = g(x) * h(x)$.

- Homogeneidad $f(x) * a_1 g(x) = a_1 (h(x) * g(x))$ con $a_1 \in \mathbb{R}$.
- Distributividad $g(x) * (h_1(x) + h_2(x)) = g(x) * h_1(x) + g(x) * h_2(x)$.
- Asociatividad $g(x) * (h_1(x) * h_2(x)) = (g(x) * h_1(x)) * h_2(x)$.
- Amplificación $g(x) * a_1 \delta(x) = a_1 g(x)$ con $a_1 \in \mathbb{R}$.
- Desplazamiento $g(x) * \delta(x + s) = g(x + s)$
- Eco: $g(x) * (\delta(x) + k\delta(x - s)) = g(x) + k g(x - s)$.

4.5 Algunas convoluciones importantes: Aproximación discreta de la primer derivada

Tome la siguiente convolución discreta:

$$h[n] = g[n] * f[n],$$

con $g[n] = \begin{pmatrix} 1 & -1 \end{pmatrix}$ y $f[n]$ cualquier función, examinemos por ejemplo los valores $h[2]$ y $h[3]$:

$$h[2] = \sum_{m=1}^2 f[m] g[2-m] = f[1] g[1] + f[2] g[0] = f[2] - f[1]$$

$$h[3] = \sum_{m=2}^3 f[m] g[3-m] = f[2] g[1] + f[3] g[0] = f[3] - f[2]$$

Generalizando lo anterior, se concluye que:

$$d_-[n] = f[n] - f[n-1] \quad (4)$$

y haciendo el reemplazo de $n = u + 1$ se obtiene lo que llamaremos la aproximación por diferencias finitas $d_+[u]$:

$$d_+[u] = f[u+1] - f[u] \quad (5)$$

Veamos ahora la definición formal de la derivada de una función continua $f(x)$, la cual mide la tasa de cambio del funcional $f(x)$ en un punto x (tomando en cuenta la dirección del mismo):

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}.$$

Aproximación de la derivada por diferencias finitas: basados en la derivada según la definición continua, para el dominio discreto, hacemos que h tenga un valor discreto menor posible, el cual corresponde a $h = 1$, por lo que entonces se tiene que, a partir de la definición formal de la primera derivada:

$$f'[n] \cong d_+[n] = \frac{f[n+h] - f[n]}{h} = f[n+1] - f[n].$$

La fórmula anterior se denomina **aproximación por diferencias hacia adelante**, pues involucra un desplazamiento hacia adelante de la función y coincide con lo arribado en la ecuación 5. La aproximación por **diferencias hacia atrás** se define entonces como:

$$f' [n] \cong d_- [n] = \frac{f [n] - f [n - h]}{h} = f [n] - f [n - 1],$$

lo cual coincide con lo planteado en la ecuación 4. Una mejor aproximación es la de las **diferencias centrales**, con un error menor y menor sensibilidad al ruido, la cual promedia las aproximaciones hacia adelante y atrás:

$$f' [n] \cong \frac{1}{2} \left(\frac{f [n + h] - f [n]}{h} + \frac{f [n] - f [n - h]}{h} \right) = \frac{f [n + h] - f [n - h]}{2h} = \frac{f [n + 1] - f [n - 1]}{2} \quad (6)$$

La última igualdad, considerando a $h = 1$. Observe entonces que la convolución de una función $f [n]$ con una **máscara** o **núcleo** (*kernel* en inglés) $d_+ [n] = (1 \quad -1)$ resulta en una aproximación de la primer derivada por diferencias hacia adelante. Para realizar una aproximación por diferencias centrales, según lo fijado en la ecuación 6 se usa la máscara:

$$d_c [n] = \left(\frac{1}{2} \quad 0 \quad -\frac{1}{2} \right).$$

Es usual la necesidad de amplificar la magnitud de la aproximación de la primer derivada en muchos casos, por lo que por ejemplo, por la propiedad de la homogeneidad de la convolución podemos crear una máscara nueva $d_c^* [n] = \alpha d_c [n]$, por lo que con $\alpha = 2$, se obtiene la máscara:

$$d_c^* [n] = (1 \quad 0 \quad -1).$$

Por ello entonces, la primera derivada $f' [n]$ en términos de la convolución se puede aproximar como:

$$f' [n] \cong f [n] * d_c^* [n]$$

4.6 Aproximación discreta de la segunda derivada

La segunda derivada de una función f corresponde a la derivada de la primer derivada $f' (x)$, lo que conceptualmente se refiere entonces a la *aceleración* con que cambian los valores del funcional f respecto a x . Es por ello que entonces la segunda derivada corresponde a:

$$\begin{aligned} f'' (x) &= \lim_{h \rightarrow 0} \frac{f' (x + h) - f' (x)}{h} = \lim_{h \rightarrow 0} \frac{\lim_{h \rightarrow 0} \frac{f(x+2h) - f(x+h)}{h} - \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}}{h} \\ &\Rightarrow f'' (x) = \lim_{h \rightarrow 0} \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2}, \end{aligned}$$

con lo cual tomando $h = 1$ se tiene que, en términos discretos, se puede aproximar la segunda derivada como sigue:

$$\Rightarrow f''(x) \cong f(x+2) - 2f(x+1) + f(x),$$

y su máscara correspondiente se define entonces como:

$$s[n] = (1 \quad -2 \quad 1)$$

y de manera similar se tiene entonces que, en términos de una convolución, la segunda derivada se define como:

$$f''[n] \cong f[n] * s[n] \quad (7)$$

5 Filtros lineales

5.1 Filtros paso bajos o de eliminación de ruido

Como se observó anteriormente para el caso de la función $rect \sqcap(x, y)$, su convolución con cualquier otra función $f(x, y)$ tiende a resultar en un *suavizado* de $f(x, y)$, y en tal caso visto anteriormente, la convolución $\sqcap(x, y) * \sqcap(x, y) = \wedge(x)$ resultó en la función triángulo, la cual puede conceptualizarse como una aproximación más suave de la función $rect$. La función $rect$ con dominio en dos dimensiones, puede representarse una matriz de 1's $\sqcap \in \mathbb{R}^{N \times M}$ en todas sus entradas:

$$\sqcap = \alpha \begin{bmatrix} 1 & \dots & 1_{1,M} \\ \vdots & \ddots & 1 \\ 1_{N,1} & \dots & 1_{N,M} \end{bmatrix}$$

con $\alpha = \frac{1}{NM}$ el coeficiente de normalización que se asegura de preservar la magnitud de la señal original, por lo que también se llama el filtro de promediado.

Visto en el dominio de Fourier (transformación $\mathcal{F}(U)$ que descompone la señal U en una sumatoria de senos y cosenos complejos), los cambios abruptos o fuertes en una señal necesitan de componentes de alta frecuencia con alta amplitud, para aproximar la función original, por lo que entonces se puede afirmar que los bordes o cambios de alta magnitud y abruptos se asocian con la energía en las altas frecuencias. La energía entonces en las bajas frecuencias, se asocian a cambios suaves en la señal. Por ejemplo, para una imagen, los bordes y el ruido en general tienden a generar energía en las altas frecuencias, e información como la textura genera la energía en intervalos de frecuencia más bajos. Es por esto que convolucionar una señal U con un \sqcap , $U * \sqcap$, resulta en una atenuación fuerte de la energía en las altas frecuencias, y menor en las bajas frecuencias, por lo que el filtro de promediado es un tipo de filtro que deja pasar la energía en las bajas frecuencias, o paso bajos.

Un filtro más utilizado aún para los mismos efectos, es el filtro Gaussiano $\mathcal{N}(x, y)$, el cual tiene la agradable propiedad de no tener discontinuidades y ser una función *suave*, distinto al filtro basado en la función *rect*, cuyas discontinuidades hacen que sea necesario una cantidad infinita de senos y cosenos para representarlo, puesto que:

$$\mathcal{F}\left(\prod(x)\right) = \text{sinc}(x)$$

lo cual introduce elementos de frecuencia no deseados. En cambio, para el caso de la función Gaussiana, su transformada es también una función Gaussiana $\mathcal{F}(\mathcal{N}(x, y)) = \mathcal{N}(x, y)$, por lo que truncarla para que tenga un dominio finito, provoca menos inexactitud, además de incluir menos energía de componentes no deseados.

Usualmente la función Gaussiana implementada para el filtrado es *isotrópica*, es decir, es equivalente sin importar la dirección desde la que se le examine. Es por esto que entonces la función viene dada entonces por:

$$F(x, y) = \mathcal{N}(x, y) = \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right)$$

Por ejemplo, utilizando la función simétrica en ambas dimensiones, con desviación estándar $\sigma_x = \sigma_y = 1$ y dimensiones $\mathcal{N} \in \mathbb{R}^{3 \times 3}$

```
fspecial('gaussian', 3, 1)
```

se obtiene la máscara (con el coeficiente de normalizado incluido):

$$\mathcal{N}[x, y] = \begin{bmatrix} 0,0751 & 0,1238 & 0,0751 \\ 0,1238 & 0,2042 & 0,1238 \\ 0,0751 & 0,1238 & 0,0751 \end{bmatrix}$$

la cual ya está normalizada de forma que la sumatoria de todos los elementos resulta en 1, e **implementa una covarianza nula entre las dos dimensiones**.

La Figura 17 ilustra la superficie generada por una función Gaussiana. Además, la Figura 17 ilustra el efecto de convolucionar una imagen con un núcleo Gaussiano.

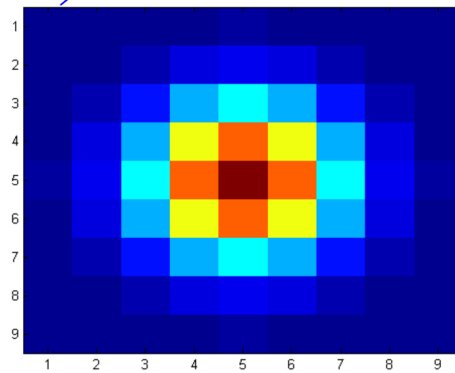
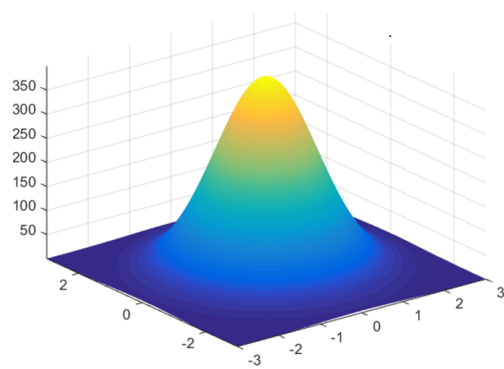
5.2 Filtros paso altos: detección y mejora de bordes

5.2.1 Filtros de aproximación al gradiente de primer grado (primera derivada)

Los filtros de Prewitt y Sobel para la detección de bordes, aproximan al gradiente para una función (matriz) $U \in \mathbb{R}^{N \times M}$ con $z = U[x, y]$ como:

$$\nabla U = \frac{d}{dx}U[x, y]\hat{i} + \frac{d}{dy}U[x, y]\hat{j} = \nabla_x U\hat{i} + \nabla_y U\hat{j}$$

Las máscaras en general utilizadas para realizar la aproximación de las derivadas parciales $\frac{d}{dx}U[x, y]$ y $\frac{d}{dy}U[x, y]$ están dadas, respectivamente como:



$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



Figura 17: Ilustraciones de la función de densidad Gaussiana en distintas perspectivas, y el resultado de convolucionar una imagen con un núcleo Gaussiano.

$$d_x = \alpha \begin{bmatrix} 1 & 0 & -1 \\ k & 0 & -k \\ 1 & 0 & -1 \end{bmatrix} \quad d_y = \alpha \begin{bmatrix} 1 & k & 1 \\ 0 & 0 & 0 \\ -1 & -k & -1 \end{bmatrix}$$

donde para el filtro de Prewitt $k = 1$ y $\alpha = \frac{1}{6}$, y para el filtro de Sobel $k = 2$ y $\alpha = \frac{1}{8}$. El coeficiente α se asegura de preservar la amplitud original de la señal para las derivadas parciales, y el coeficiente k hace más sensible el operador a los cambios en la señal. De esta manera, las derivadas parciales se aproximan según Prewitt y Sobel como:

$$\frac{d}{dx}U \approx U * d_x \quad \frac{d}{dy}U \approx U * d_y.$$

Para la detección de bordes, es suficiente conocer la magnitud del gradiente en cada punto, por lo que entonces se puede aproximar el mismo como:

$$\|\nabla U(x, y)\| = \sqrt{\left(\frac{d}{dx}U[x, y]\right)^2 + \left(\frac{d}{dy}U[x, y]\right)^2}$$

Una umbralización automática, usando por ejemplo el algoritmo de Otsu, permite detectar los bordes a partir de la matriz de magnitudes de gradientes $\|\nabla f(x, y)\|$. Un problema notable de la aproximación de la magnitud del gradiente es su sensibilidad al ruido, por lo que para el caso de la detección de bordes, se utilizan los cruces por cero de la segunda derivada. La Figura 18 muestra como la aproximación de la segunda derivada facilita la detección de bordes, usando el concepto de cruces por cero.

5.2.2 Filtros de aproximación al gradiente de segundo grado (segunda derivada), el Laplaciano del Gaussiano (LoG)

Ya anteriormente se demostró que la convolución de una función $u(t)$, $u : \mathbb{R} \rightarrow \mathbb{R}$ con una máscara $s[n] = (1 \ -2 \ 1)$ aproxima a la segunda derivada. La extensión de esta máscara como una función $L(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$ para aproximar la segunda derivada de se puede definir como:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -k & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

con $k = 4$ o $k = 8$ por ejemplo, la máscara aproxima primero la segunda derivada respecto a x , y luego respecto a y , en una sola máscara:

$$(U * L)[x, y] = \nabla_{x,y}U[x, y].$$

La Figura 18 muestra la aproximación a la segunda derivada. Para mejorar la robustez ante el ruido, es común convolucionar la imagen de entrada U con un

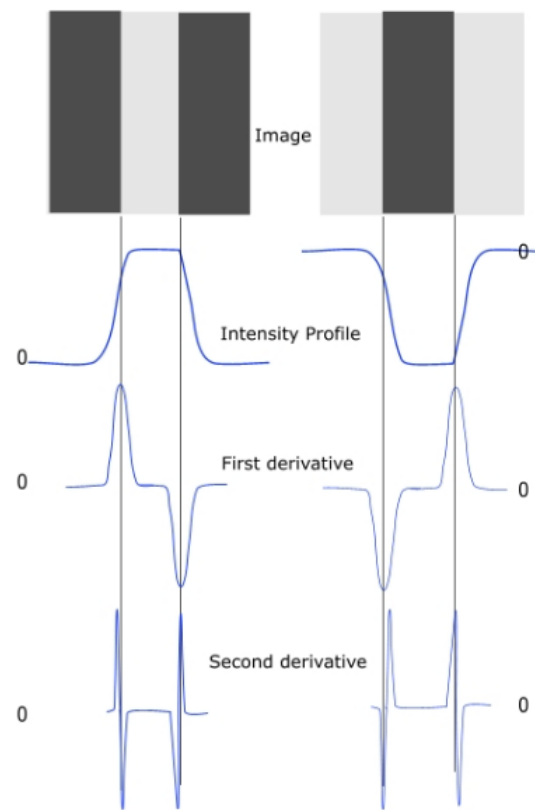


Figura 18: Aproximación de la primera y segunda derivada.

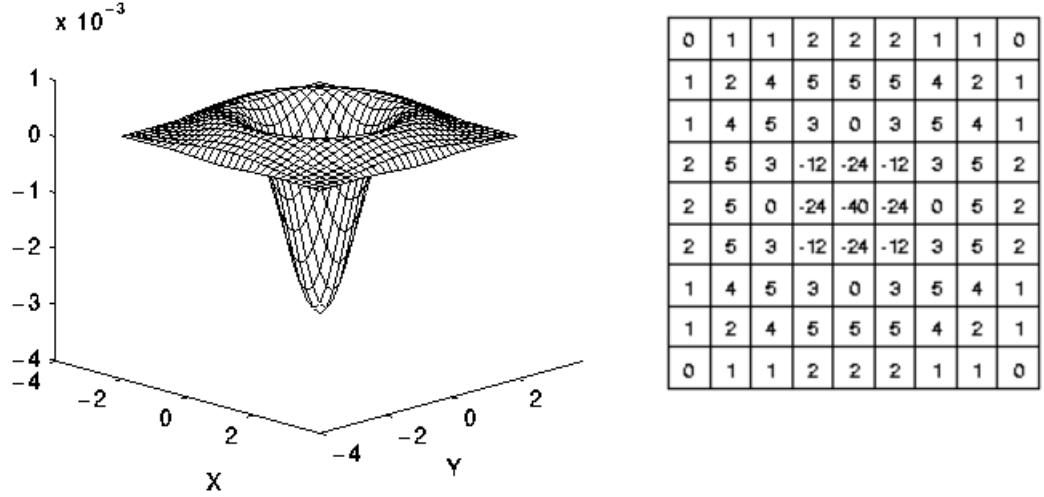


Figura 19: Laplaciano del Gaussiano, ejemplo. Tomado de <https://www.quora.com/What-happens-to-the-image-when-Laplacian-of-Gaussian-is-applied>

núcleo Gaussiano \mathcal{N} , para posteriormente convolucionar la matriz resultante por la máscara Gaussiana L , de modo que

$$((U * \mathcal{N}) * L)[x, y]$$

, lo que supone dos convoluciones o deslizamientos sobre la imagen $U \in \mathbb{R}^{N \times M}$. Para ahorrar costo computacional, utilizamos las propiedades de la conmutatividad y la asociatividad, de modo que

$$((U * \mathcal{N}) * L)[x, y] = (U * (L * \mathcal{N}))[x, y],$$

lo que resulta en una nueva máscara conocida como el Laplaciano del Gaussiano, o LoG:

$$\text{LoG}[x, y] = (L * \mathcal{N})[x, y],$$

observe que los coeficientes de la máscara LoG dependen de las desviaciones estándar usadas en la máscara Gaussiana σ_x y σ_y . La Figura 19 muestra gráficamente una máscara LoG de ejemplo.

La convolución de la imagen de entrada U con el núcleo LoG resulta en una imagen

$$E = U * \text{LoG}$$

con cruces por cero donde existan bordes, como ilustra la Figura 20, la cual fue normalizada en el rango de 0-255 para evitar la representación de los valores negativos.

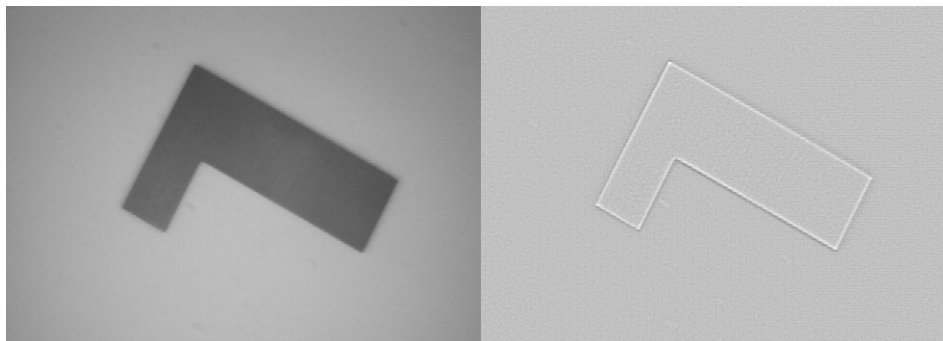


Figura 20: Visualización de la imagen $E = U * \text{LoG}$.

Una forma alternativa de calcular la respuesta al filtro LoG E es utilizando lo que se conoce como la diferencia entre Gaussianas (DoG), la cual consiste en tomar la diferencia entre la imagen original U y el resultado de convolucionar tal imagen con un núcleo Gaussiano \mathcal{N} , por lo que entonces se tiene que:

$$E \approx U - (U * \mathcal{N})$$

a la imagen $B = -E$ se le conoce como la máscara deforme o *unsharp mask*. La Figura 21 ilustra el proceso de aproximación de la segunda derivada usando DoG, y el proceso que sucede al observar la gráfica de un borde.

5.2.3 Mejora de bordes: *Unsharp Masking*

Para la mejora de bordes, se puede usar la máscara deforme o *unsharp mask* definida anteriormente como $B = -E$ y super-ponerla para producir la imagen con bordes mejorados G :

$$G = U + \lambda B$$

donde $\lambda \in \mathbb{R}$ es el coeficiente que controla la mejora de los bordes. La Figura 22 ilustra el proceso de mejora de bordes definido anteriormente, y conocido también como *unsharped masking*.

6 Filtros no lineales

Los filtros no lineales permiten generar comportamientos más sofisticados, y evitar problemas como por ejemplo la degradación de bordes en el caso de los filtros paso bajos, o la magnificación del ruido en los filtros paso altos. A continuación se presentan algunos filtros no lineales para distintos propósitos.

6.1 Filtro de medianas

El filtro de medianas es un filtro no lineal, el cual toma la mediana de cada ventana. La mediana se define como el valor a la *mitad* de un conjunto de datos

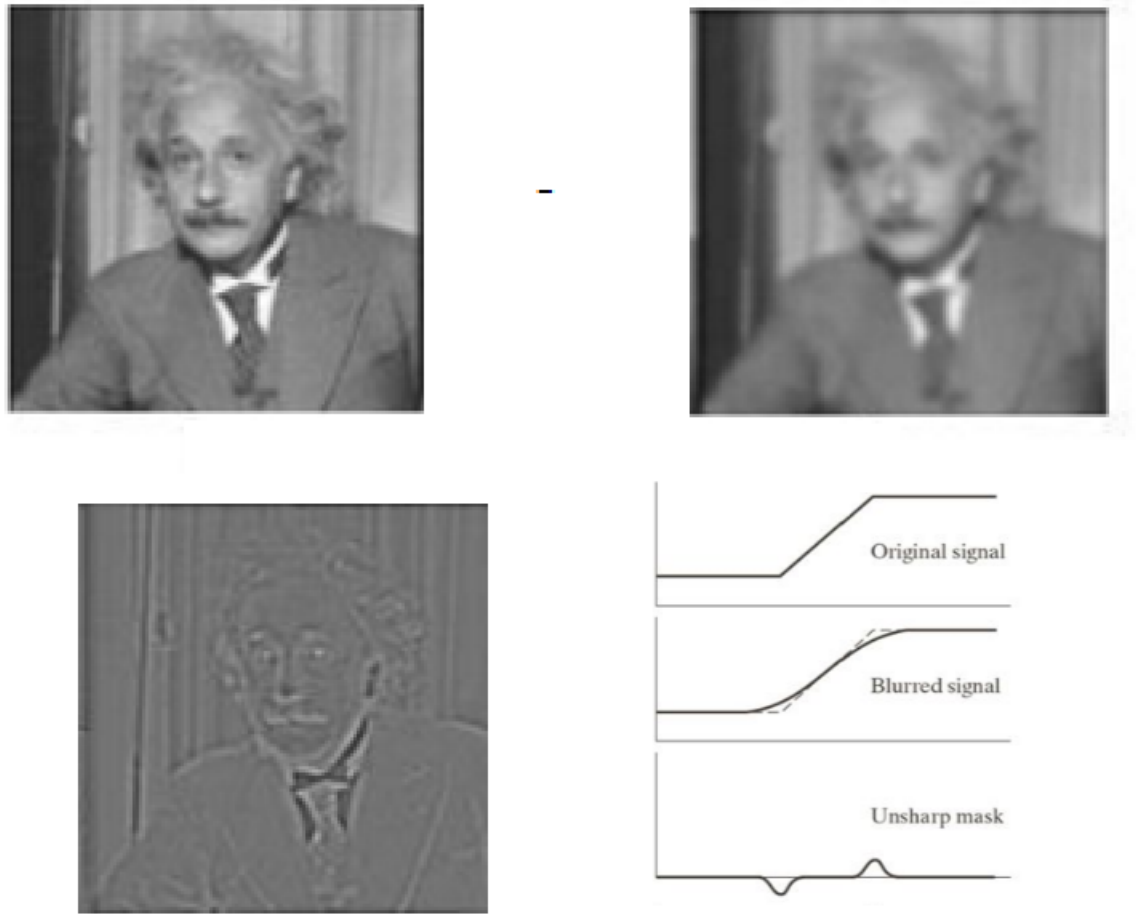


Figura 21: Ilustración de DoG. Tomado de http://ict.udlap.mx/people/oleg/docencia/imagenes/chapter3/image_32_IS548.html

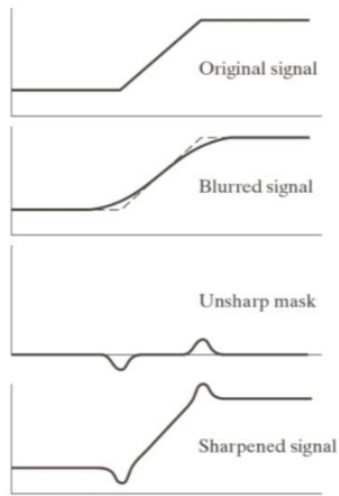


Figura 22: Unsharped Masking con distintos valores de λ . Tomado de http://ict.udlap.mx/people/oleg/docencia/imagenes/chapter3/image_32_IS548.html



Figura 23: Imagen contaminada con ruido de tipo sal y pimienta y el resultado de filtrarla con un filtro de medianas.

ordenados. Por ejemplo, para la siguientes matriz $A \in \mathbb{R}^{N \times N}$, con en este caso $N = 3$:

$$A = \begin{bmatrix} 3 & 2 & 4 \\ 5 & 7 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

al calcular un vector \vec{a} con todos los elementos ordenados de mayor a menor se tiene que:

$$\vec{a} = [0 \quad 1 \quad 1 \quad 2 \quad \mathbf{2} \quad 3 \quad 4 \quad 5 \quad 7]$$

el elemento a la mitad o en la mediana del arreglo ordenado, en la posición $\frac{N \times N - 1}{2}$ (indizado desde 0) de los elementos de A viene dado por:

$$\text{mediana}(A) = 2.$$

El filtro de medianas define con una ventana de $K \times L$ dimensiones, realiza el cálculo de la mediana para esa ventana deslizante en una imagen de entrada $U \in \mathbb{R}^{N \times M}$:

$$G[x, y] = \text{mediana} \left(U_{x - \frac{(K-1)}{2} : x + \frac{(K-1)}{2}, y - \frac{(L-1)}{2} : y + \frac{(L-1)}{2}} \right), \quad (8)$$

El filtro de medianas es un filtro de la familia de filtros de *orden de rango* o *rank order filters*, los cuales obtienen resultados satisfactorios en señales contaminadas con ruido de tipo *sal y pimienta* como el ilustrado en la Figura 23.

6.2 Filtro bilateral

El filtro bilateral, además de pesar el dominio espacial con una función Gaussiana, pesa el dominio de la intensidad con otra función Gaussiana, de modo que los píxeles muy distintos al píxel en el que está centrado la ventana deslizante, obtienen un peso más bajo, evitando la degradación de los bordes.

El filtro bilateral [?] implementa una función dependiente de la entrada U :

$$S_{\text{BF}}[x, y] = \frac{1}{\alpha} \sum_{u=0}^{K-1} \sum_{v=0}^{L-1} U \left[x + u - \left(\frac{K-1}{2} \right), y + v - \left(\frac{L-1}{2} \right) \right] \psi_{\text{BF}}[[u, v], [x, y]]. \quad (9)$$

$$\psi_{\text{BF}}[[u, v], [x, y]] = \exp \left(-\frac{((u-x)^2 + (v-y)^2)}{2\sigma_s^2} \right) \exp \left(-\frac{(U[x, y] - U[u, v])^2}{2\sigma_r^2} \right).$$

Como se puede observar, se definen dos parámetros, σ_r y σ_s , los cuales controlan el ancho de la campana Gaussiana en el dominio de la intensidad y del espacio, respectivamente. La Figura 24 muestra el efecto de cada parámetro.

Analizando el comportamiento del Filtro bilateral matemáticamente, y según la Figura 24, se observa que si $\sigma_s \rightarrow \infty$:

$$F[x, y] = \exp \left(-\frac{((u-x)^2 + (v-y)^2)}{2\sigma_s^2} \right) \exp \left(-\frac{(U[x, y] - U[u, v])^2}{2\sigma_r^2} \right) = \exp \left(-\frac{(U[x, y] - U[u, v])^2}{2\sigma_r^2} \right),$$

el filtro pasa a usar una ventana rectangular ideal, pues no existe un pesado en el dominio espacial. Lo anterior simplifica el cómputo, pero conlleva el efecto de *aliasing* y difuminación sensible a la dirección (borroneado cuadrado).

Respecto al efecto de hacer $\sigma_r \rightarrow \infty$ se tiene que:

$$F[x, y] = \exp \left(-\frac{((u-x)^2 + (v-y)^2)}{2\sigma_s^2} \right) \exp \left(-\frac{(U[x, y] - U[u, v])^2}{2\sigma_r^2} \right) = \exp \left(-\frac{((u-x)^2 + (v-y)^2)}{2\sigma_s^2} \right),$$

por lo que el filtro se comporta como un filtro Gaussiano, el cual no toma en cuenta la intensidad de la imagen, por lo que los bordes se degradan.

6.3 Filtro de promediado no local

Ante situaciones de mucho ruido (una relación señal a ruido muy baja) el filtro bilateral no presenta un buen desempeño. El Filtro de promediado no local (Non Local Means o NLM) utiliza la premisa de pesar no solo la similitud de intensidad entre dos píxeles $p_1 = (x_1, y_1)$ y $p_2 = (x_2, y_2)$, sino también la similitud de los vecindarios de ambos píxeles.

La Figura 25 muestra a los píxeles q_1 , q_2 y q_3 , además de los vecindarios de cada uno en el recuadro blanco de $M \times M$ píxeles, donde en tal caso los píxeles q_1 y q_2 son bastante más similares entre sí que los píxeles q_3 y q_1 . De esta forma se minimiza la incorrecta interpretación que hace el filtro Bilateral de comparar las intensidades de dos píxeles p_1 y p_2 que sean similares en cuanto

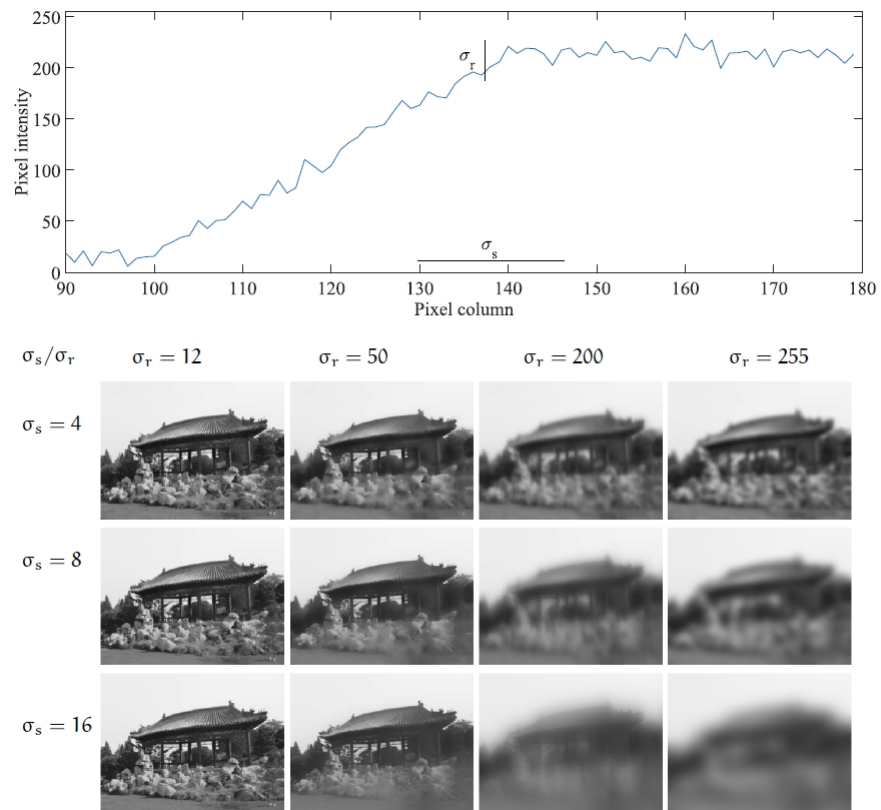


Figura 24: Efectos de los dos parámetros del filtro bilateral, tomado de P. Kornprobst and J. Tumblin, *Bilateral filtering: Theory and applications*. Now Publishers Inc, 2009..

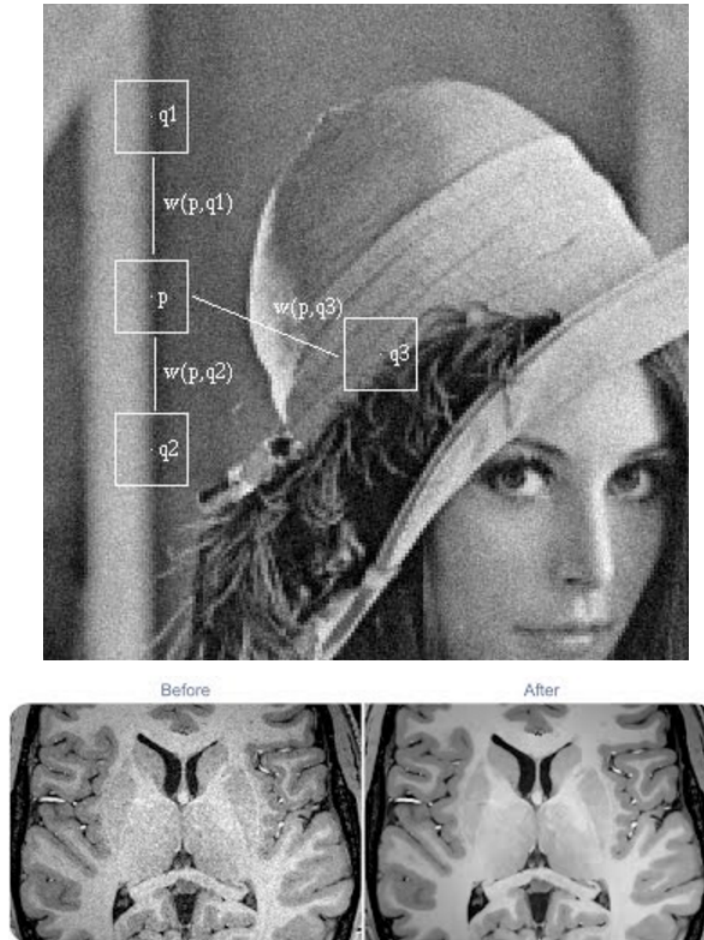


Figura 25: Filtro de promediado no local, tomado de *A non-local algorithm for image denoising*.

a intensidad por causa del ruido (en cuyo caso usualmente no existe una similitud estructural de sus vecindarios). En tal caso, el filtro NLM resultaría en un peso bajo al comparar los vecindarios de los píxeles p_1 y p_2 .

Para asignar la similitud de los vecindarios entre píxeles, el NLM vectoriza los vecindarios de ambos píxeles en $\vec{\eta}(p_1)$ y $\vec{\eta}(p_2)$, respectivamente, y realiza la comparación de tales vecindarios usando por ejemplo la norma euclidiana $\|\vec{\eta}(p_1) - \vec{\eta}(p_2)\|_2$. De esta forma, la ecuación del filtro se expresa como:

$$F[x, y] = \exp\left(-\frac{(x^2 + y^2)}{2\sigma_s^2}\right) \exp\left(-\frac{\|\vec{\eta}(p_1) - \vec{\eta}(p_2)\|_2}{2\sigma_r^2}\right).$$

$$\psi_{\text{NLM}}[[u, v], [x, y]] = \exp\left(-\frac{((u - x)^2 + (v - y)^2)}{2\sigma_s^2}\right) \exp\left(-\frac{\|\vec{\eta}[x, y] - \vec{\eta}[u, v]\|}{2\sigma_r^2}\right).$$

Una implementación a *fuerza bruta* del NLM es considerablemente más costosa que por ejemplo una implementación del mismo tipo del filtro bilateral. Es por eso que la implementación de este tipo usual del NLM elimina el término de pesado espacial, usando una ventana rectangular:

$$F[x, y] = \exp\left(-\frac{\|\vec{\eta}[x, y] - \vec{\eta}[u, v]\|}{2\sigma_r^2}\right).$$

Se pueden además considerar otras distancias más rápidas de calcular, como una del tipo ℓ_∞ .

7 Métricas para la eliminación del ruido y exactitud de bordes

7.1 Métricas para la eliminación del ruido: Relación señal a ruido

Existen distintas métricas para medir la cantidad de ruido presente en una señal. La relación pico señal-ruido (*peak signal to noise ratio*) es muy utilizada en Ingeniería Eléctrica, y la medición de las ondas sonoras. Se basa en el logaritmo de base 10 para representar la relación entre la potencia eléctrica y la potencia de la señal (para bajas potencias eléctricas, un cambio tiene mayor efecto en la potencia de la señal que en altas potencias eléctricas). Su fórmula viene dada por:

$$\text{PSNR} = 10 \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) = 20 \log_{10}\left(\frac{\text{MAX}}{\sqrt{\text{MSE}}}\right) = 20 \log_{10}(\text{MAX}) - 10 \log_{10}(\text{MSE}),$$

donde MAX se refiere al máximo valor posible de la señal, por ejemplo, para imágenes a escala de grises $\text{MAX} = 255$ y el error medio cuadrático (*mean*

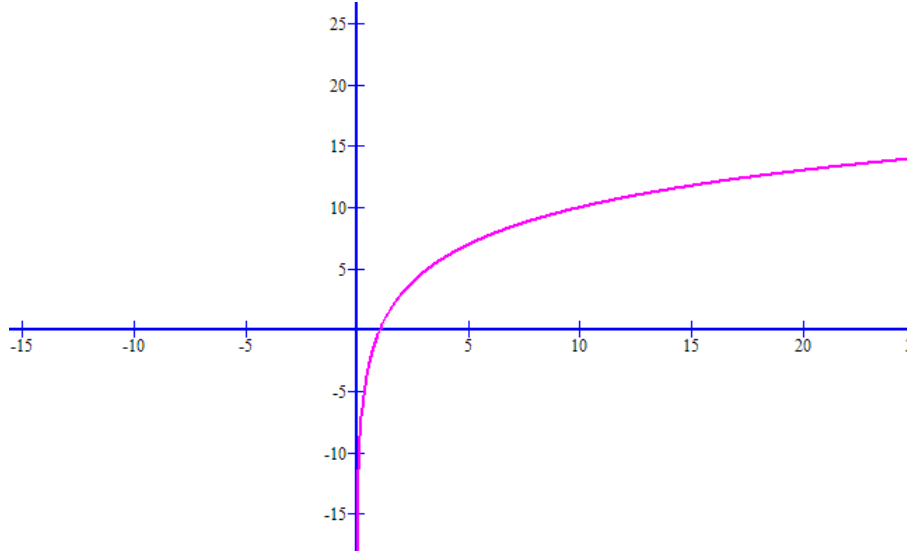


Figura 26: Graficación de la función \log_{10} .

squared error) de la diferencia entre la señal original, sin contaminar por ruido $U \in \mathbb{R}^{m \times n}$ (la imagen de entrada), y la imagen Y contaminada por ruido:

$$\text{MSE} = \frac{1}{m n} \sum_{i=1}^m \sum_{j=1}^n (U(i, j) - Y(i, j))^2.$$

La relación $\frac{\text{MAX}^2}{\text{MSE}}$ asegura que el argumento del logaritmo sea siempre mayor que 1, pues el máximo error promedio puede llegar a ser MAX^2 . El logaritmo multiplicado por 10 (de allí el prefijo *deci*) se comporta como lo mostrado en la Figura 26, donde se observa que para bajos valores, un cambio supone una mayor ganancia, y en altos valores, un cambio supone una ganancia menor.

Observe que en caso de existir un alto error medio cuadrático, relación señal a ruido será más baja, de lo contrario, en presencia de poco ruido, la relación señal a ruido es bastante más alta, como lo ilustra la Figura 27.

7.2 Métricas para la comparación entre señales binarias

Para evaluar la precisión de la segmentación, las métricas usuales evalúan la diferencia de señales binarias (donde por ejemplo los píxeles etiquetados con 1 se refieren a los objetos de interés y los etiquetados con 0 al fondo). Una métrica muy utilizada para tales efectos es el coeficiente de *dice* o de Sørensen, el cual se puede usar para evaluar la similitud estructural entre dos imágenes binarias

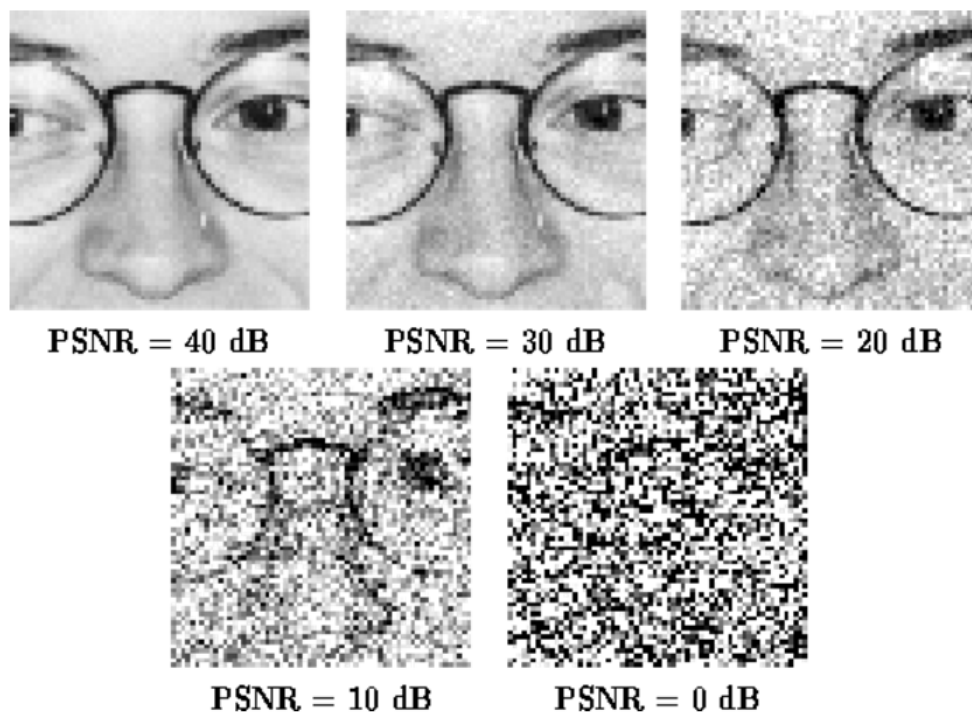


Figura 27: PSNR de distintos valores. Tomado de http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/VELDHUIZEN/node18.html



Figura 28: *Groundtruth* a la izquierda, e imagen automáticamente segmentada a la derecha. Tomado de <https://blog.altoros.com/experimenting-with-deep-neural-networks-for-x-ray-image-segmentation.html>

$U \in \mathbb{R}^{m \times n}$ y $V \in \mathbb{R}^{m \times n}$, y viene dado por:

$$S = \frac{2 \left(\sum_j^m \sum_i^n U[i, j] V[i, j] \right)}{\left(\sum_j^m \sum_i^n U[i, j] \right) + \left(\sum_j^m \sum_i^n V[i, j] \right)}$$

donde el numerador equivale conceptualmente a la cantidad de pixeles del objeto de interés intersecados en ambas imágenes U y V : $2|U \cap V|$ (donde las barras denotan la cantidad de pixeles con 1's en la matriz) y el denominador a la suma de los pixeles del objeto de interés en ambas imágenes: $|U| + |V|$, lo que entonces significa que se puede reescribir S como:

$$S = \frac{2|U \cap V|}{|U| + |V|}$$

La imagen U puede asociarse a la imagen segmentada por el algoritmo, y V la imagen segmentada manualmente, también conocida como imagen de *groundtruth*. De este modo, una segmentación perfecta según el *groundtruth* resulta en un coeficiente de *dice* $S = 1$, y una segmentación completamente errada un coeficiente $S = 0$. La Figura 28 ilustra la segmentación automática de una radiografía pulmonar, y el *groundtruth* de la misma.

Referencias

- [1] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins. *Digital image processing using MATLAB*. Pearson Education India, 2004.
- [2] Pablo Irarrázaval. *Análisis de señales*. McGraw-Hill Interamericana, 1999.